IBM Application Discovery for IBM Z V6.1.0

User Guide



SC28-3160-04

Edition notice

May 2022 edition

This edition applies to version 6, release 1, modification 0 of IBM® Application Discovery for IBM Z® (program number 5737-B16) and to all subsequent releases and modifications until otherwise indicated in new editions.

It is our intention to update the product documentation for this release periodically, without updating the form number. If you are not using the latest version of the product and need to refer to earlier versions of the product documentation, PDF versions are provided as alternatives. To download the PDF files, go to https://www.ibm.com/docs/en/addi/6.1.0?topic=download-pdf-versions.

© Copyright IBM Corp. 2010, 2022

Contents

Edition notice iii

Chapter 1. Accessibility features for IBM AD 1

Chapter 2. Product Overview 3
- IBM AD High-Level Architecture Overview 4
- Terms and Conventions 6

Chapter 3. Installing 7
- Prerequisites 7
  - CPU, RAM, and Storage Requirements 7
  - Supported Platforms and Versions 8
  - Prerequisite Software and Configurations 9
  - TCP Port Requirements and Firewall Exceptions 20
- Installing components with the IBM ADDI Installer 25
- Installing components by using CLI 27
- Installing IBM AD Analyze Client 27
- Installing IBM AD Connect for Mainframe 29

Chapter 4. Upgrading 31

Chapter 5. Configuring 33
- Configuring by using IBM AD Configuration Wizard 33
- Configuring by using IBM AD Configuration Service 35
  - Advanced Configurations 36
  - Manual Configurations 54
- Configuring IBM AD Analyze Client 149
- Configuring IBM AD Connect for Mainframe 152
  - Creating a z/OS Connection 152
  - Technical Configuration Process 152

Chapter 6. Managing Security in IBM AD 203
- Securing Apache ZooKeeper SSL connections 203
  - Setting up IBM AD ZooKeeper to run in Mixed Mode 203
  - Block the incoming traffic for the non-authenticated port of IBM AD Zookeeper 205
- Enabling encryption channel between IBM AD Build Client and IBM AD ZooKeeper 206
- Enabling encryption channel between IBM AD Analyze Client and IBM AD ZooKeeper 208
- Obtaining an encrypted keystore password 209
- Other security recommendations 210
  - IBM AD Build Client - Enable SMB2 Encryption to avoid any possible sniffing of the network packages 210
  - IBM AD Build Client - DLL Hijacking 210
  - Securing Access on IBM AD Services Endpoints 210
  - Using the Authentication and Authorization Feature 211
  - Mask usernames in IBM AD Configuration Server 212

Chapter 7. Managing Application Projects with IBM AD Build 213
- Using IBM AD Build Configuration 213
  - Viewing Project Information 214

Deleting a Project ... 214
Renaming a Project ... 214
Recreate a Repository ... 215
Upgrade a Repository ... 215
Stop the Mainframe Import ... 215
Configuring the z/OS Connection ... 216
Bringing Operational Information ... 228
Bringing data from mainframe libraries (PDS Libraries, Endevor, Librarian, Natural) ... 233
Bringing Data From Mainframe Using ChangeMan® ZMF ... 235
The zOS Tab ... 236
Automatic Messaging ... 238
Enabling the Assembler Macro Expansion Feature ... 238
Enabling the z/OS User Authentication feature for IBM AD Build Configuration ... 238
Using IBM AD Build Client ... 240
UI Reference ... 241
Tasks ... 247
IBM AD Build CLI Commands ... 297
I. Overview ... 297
II. Description of the IBM AD Build Client Batch Commands ... 297
III. Description of the IBM AD Build Configuration Batch Commands ... 302
IV. Use Cases and Best Practices ... 303
V. Setting up Automatic Updates with Windows Scheduler ... 305

Chapter 8. Analyzing Applications with IBM AD Analyze ... 307
UI Reference ... 307
Perspectives ... 307
Graphs ... 330
Reports ... 357
Usage ... 388
Compare Analysis ... 424
Tabs in IBM AD Analyze Server Manager ... 425
Customizing Preferences ... 429
Environment Settings ... 429
Authorization and Authentication ... 430
Local Settings ... 431
Creating a Mainframe Project ... 447
Creating a Legacy AD Java Project ... 447
Importing a Shared Project from AD Analyze Server ... 452
Checking the Connection to the Project Database ... 454
Building a Legacy AD Java Project ... 455
Analyzing a Project ... 456
Handling Annotations ... 456
Annotations Overview ... 456
Adding an Annotation ... 457
View Annotations ... 458

Chapter 9. Extended Capabilities ... 461
Activating the full functionality of IBM AD from a free version ... 461
Integrating with IBM License Metric Tool ... 461
Integrating IBM AD Analyze Client with IDz features ... 462
z/OS Subsystem and Third-Party Product Configuration Checklists ... 464
Analyzing Resources From an Endevor Remote System ... 464
Analyzing resources through APIs ... 465
Generating call chain analysis results by using API URLs ... 465
Handling annotations by using API URLs ... 472
EWM to AD Integration ... 474
Introduction ... 474

EWM Overview....474
Source Management on EWM....476
z/OS Specific EWM Objects....477
RTC to AD Integration Tool - RTC2AD....477
Installing and Configuring the RTC2AD Tool....478
Running the RTC2AD Tool....478
Full Load Versus Incremental Load....479
Configuration File Keyword Refence....479
Processing Summary....482

Chapter 10. Tutorials....485
API Extensibility Tutorial....485
API Extensibility Sample Files....485
Setting Up a Build with Sample Files....486
Extending from Sample Files to Your Projects....488
Managing workspaces' access rights....489
Scenario 1: All Users Can See All Projects....489
Scenario 2: Users Cannot See Any Projects....489
Scenario 3: User with Access to a Workspace Can See All Projects from It....489
Scenario 4: User with Access to a Workspace Can See Only Projects from That Workspace....490
Scenario 5: Users with Access to Specific Workspaces Can See Only Projects from their Assigned Workspace....490

Chapter 11. Samples....491
Invoke an IBM utility program from JCL that calls Db2 related applications....491
Rules File Example for Rule Based Analysis....491
Log Configuration Files Example....493
Synchronize Members Configuration File Examples....494
Extensibility JSON/Configuration File Examples....497
Preprocessing Extensibility Examples....497
API/Macro Call Extensibility Examples....500
JCL Call Extensibility Examples....513
Dependency Extensibility Examples....518
IBM AD REST API Example....521
Sample JCL and PROCS for IBM AD Connect for Mainframe....525

Chapter 12. Troubleshooting....527
Known issues and workarounds....527
Messages....528
IBM AD Connect for Mainframe Messages....528
IBM AD Build Client Messages....537
Finding Log Files....537
Cleaning up the OrientDB database....540
Disaster Recovery....540
Backing Up Steps for Components....540
Restoring Steps for Components....549

Chapter 13. Uninstalling IBM AD....555

Documentation Notices....557
Trademarks....558

Chapter 1. Accessibility features for IBM AD

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

Overview

IBM Application Discovery for IBM Z includes the following major accessibility features:

- Keyboard-only operation
- Operations that use a screen reader

IBM Application Discovery for IBM Z uses the latest W3C Standard, WAI-ARIA 1.0 (www.w3.org/TR/wai-aria/), to ensure compliance with US Section 508 (www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards) and Web Content Accessibility Guidelines (WCAG) 2.0 (www.w3.org/TR/WCAG20/). To take advantage of accessibility features, use the latest release of your screen reader and the latest web browser that is supported by IBM Application Discovery for IBM Z.

Keyboard navigation

This product uses standard navigation keys.

Interface information

For alternative installation using Command Line Installation (CLI), refer to topic “Installing components by using CLI” on page 27.

The IBM Application Discovery for IBM Z user interfaces do not have content that flashes 2 - 55 times per second.

The IBM Application Discovery for IBM Z web user interface relies on cascading style sheets to render content properly and to provide a usable experience. The application provides an equivalent way for low-vision users to use system display settings, including high-contrast mode. You can control font size by using the device or web browser settings.

The IBM Application Discovery for IBM Z web user interface includes WAI-ARIA navigational landmarks that you can use to quickly navigate to functional areas in the application.

Related accessibility information

In addition to standard IBM help desk and support websites, IBM has a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

TTY service
800-IBM-3383 (800-426-3383)
(within North America)

For more information about the commitment that IBM has to accessibility, see IBM Accessibility (www.ibm.com/able).

© Copyright IBM Corp. 2010, 2022

Chapter 2. Product Overview

This section provides a high-level description of IBM AD components.

Background

Much of the world's software is legacy code that, for the most part, is stable and reliable and still supports the core business functions for which it was originally developed. Insofar as these programs refer to business-related processing, they are still relevant but require change in accordance with ongoing requirements.

Moreover, with the development of the Internet, e-commerce, and new hardware and operating systems, the need has arisen to modernize business applications by implementing new interface standards, online transaction processing over the Web, and improved database handling methods. Some legacy systems need to be replaced by new applications that use the latest design methodologies and languages.

To support the activities described above, software tools have been developed which are able to analyze the legacy system and aid in maintaining existing programs. At the same time these tools provide critical information needed in order to generate new programs that are architecturally sound and which utilize modern design methods. Furthermore, they capitalize on the fact that, for the most part, the business processing functions of the legacy system - the business rules - will still be valid for the new application.

IBM Application Discovery for IBM Z Analyze

IBM Application Discovery for IBM Z Analyze (IBM AD Analyze) is an application portfolio management product for legacy applications, which provides understanding capabilities, impact analysis, control and data flow analysis, graphical and alphanumeric reports along with a database repository. Its main functions include the following:

- Analysis of all components of a system to determine its structure and the relationships between, among and within program entities and the system and its components. The results of this analysis are stored in a data repository and can be examined by means of graphical displays and textual reports. Data in this repository serves as a basis for further analysis and synthesis.
- Mining of the code to discover business rules of the application.

The standard edition of IBM AD Analyze can process projects developed in the following languages: zOS Cobol, DT Cobol, Natural, PL/I, ADS, ADS/O, Assembler, JAVA®.

IBM AD Analyze has a client component and a server component.

IBM Application Discovery for IBM Z Build Client

IBM Application Discovery for IBM Z Build Client (IBM AD Build Client) is an application-oriented configuration management database (CMDB) that automates application understanding and technical documentation for use in all application management activities. Synchronizing with your source configuration management system, it contains a full inventory of your application components and their details. IBM AD Build Client is an indispensable tool for support activities and a precursor to undertaking enhancements and modifications. It is designed for use by all technical staff, having management components for transparency into application metrics.

IBM AD Build Client uses a relational database as a repository for storing data. If you want to view this data, you need the relational database. The tables and fields in the repository are described in detail in *IBM Application Discovery for IBM Z Repository* document, which is provided upon request by IBM Support.

IBM AD Build Client supports the following source components:

- **OS** - z/OS®/OS-390z/OS
- **Languages** - zOS Cobol, DT Cobol, Natural, PL/I, ADS, ADS/O, Assembler

© Copyright IBM Corp. 2010, 2022

- **Databases** - Db2®, Adabas, IMS DB, IDMS, Relational, Datacom
- **Transaction Monitor** - CICS®, IMS TM
- **Mapping Types** - BMS, MFS, NLM, ADS Map
- **Batch Components** - JCL, Proc, Control Files (PARM)

IBM Application Discovery for IBM Z Connect for Mainframe

IBM Application Discovery for IBM Z Connect for Mainframe (IBM AD Connect for Mainframe) provides read-only access to mainframe resources needed to deliver information to IBM AD Build (which runs on Windows) to perform analysis.

IBM AD Connect for Mainframe Listener waits on *idle* and acts only when a request comes in from the IBM AD Build. Once the requested output is sent successfully, it returns to the *idle* state. Examples of the types of queries made by IBM AD Connect for Mainframe Listener to deliver the requested types of information to IBM AD Build include:

1. Mainframe Infrastructure, via operator commands.
2. Batch Schedule Data from CA Workload Automation CA 7 or IBM Tivoli® Workload Scheduler via a **`job list`** command.
3. Libraries, Members and Source Code from CA Endevor® (via Endevor API) or ChangeMan ZMF.
4. Source Members from PDS Libraries and Data Files from VTOC.
5. Db2 Information from SYSIBM tables including keys, triggers, packages, plans.
6. CICS Information using DFHCSDUP including lists, files, maps, transactions.
7. SMF Performance Data directly from a library.
8. MQ Information using MQ commands including queue managers, queues and channels.
9. Information from Adabas using Adabas utility.
10. Information from Natural using Natural utility.
11. Information from IMS.

Note: Starting with V5.0.3, IBM AD Connect for Mainframe is being distributed in the SMP/E format. As a result, the EZL prefix has been changed to IAY. Any instance of EZL should be assumed to be IAY.

IBM AD High-Level Architecture Overview

The following diagram illustrates IBM Application Discovery for IBM Z high-level architecture and the relationships among the different components of the suite.

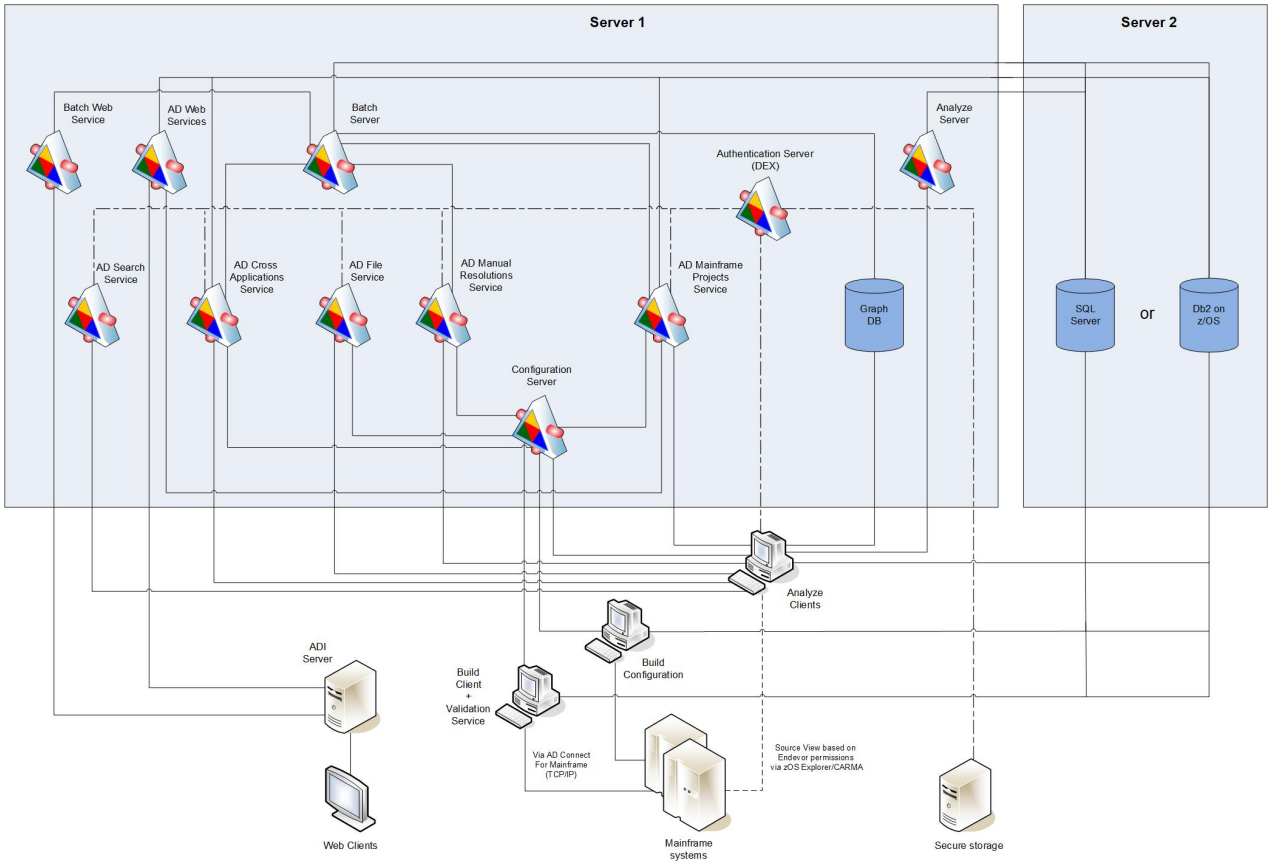


*Figure 1. **IBM AD** high-level architecture*

Following is a brief description of the relationships among the different components of IBM AD.

IBM AD Configuration Server ensures the consistency of the installation parameters throughout an installation and allows the system administrator to manage user access to workspaces.

IBM AD Build - uses data from mainframe systems to build projects. It uses project sources that are brought from z/OS®, performs a compilation/build process and stores the analysis data to the repository.

IBM AD Validation Service - works with ChangeMan SCM only. Provides coding rule enforcement via synchronization with ChangeMan and upon member staging.

IBM AD Batch Server - imports data from the relational database repository into the GraphDB (OrientDB) repository, automates processes such as report generation and indexing, and manages the creation of the annotations database.

IBM AD Analyze Client - runs over Eclipse or IDz and provides project analysis via graphs reports and usage views. When the analyzed application sources are coming from Endevor, it allows viewing source code per user based on Endevor permissions that are checked via z/OS Explorer/CARMA interface.

IBM AD Mainframe Projects - authorizes the access to the AD projects, by using SSO authentication within AD. This service is mandatory to be configured to use AD, whether the authentication feature is in place or not.

IBM AD File Service - in the context of the authorization/authentication, the access rights of users or users' groups are mapped to a certain folder that contains the source files. Once authenticated and authorized, the user can start the analysis on the source files as long as the user has read access rights.

IBM AD Search Service - is responsible with the access to the indexed data. Whether the authentication feature is in place or not, the folder path in which the indexes are generated needs to be accessible for both **IBM AD Batch Server** and **IBM AD Search Service**.

IBM AD Manual Resolutions Service - manages the manually added resolutions and allows clients that use SSO authentication within AD to ask for user authentication to access these resolutions. This service is mandatory to be configured if you want to use callgraph-based analyzes (graphs or reports), whether authentication feature is in place or not.

Authentication Server (DEX) - is an identity service that uses OpenID Connect and supports OAuth2 protocol in order to allow clients to use SSO authentication within AD. With the credentials provided by the user, it interrogates a **Secure Storage**, through the LDAP protocol. The **Secure Storage** can be an **Active Directory** or any other entity that stores users and groups and can communicate through **LDAP**. This service is mandatory to be installed and configured only when authentication feature is in place.

IBM AD Cross Applications Service - is mandatory to be configured if you want to generate a Cross Applications Callgraph, whether the authentication feature is in place or not.

Batch Web Service - serves the data that is provided by a component of the Batch Server and prepares it for delivery.

IBM AD Web Services - contains the following features: **AD Audit**, **AD Catalog** and **AD BRD REST API**.

Terms and Conventions

For the purposes of writing this guide, the following terms and conventions have been used:

- **`Command`** names are printed as shown.
- `File references` are printed as shown.
- **Button names** and options/functions within a dialog box are printed as shown.
- Throughout all examples in this guide, the underlined text will indicate where a customer can change/use their own standards or naming conventions as IBM does not mandate.

Chapter 3. Installing

Prerequisites

CPU, RAM, and Storage Requirements

Note: For detailed system requirements, you can find a software product compatibility report at https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html.

For Production Level implementations

The minimum recommended hardware requirements for Production-level implementations with multiple users and projects:

Note:

- The dedicated database server is a server with a running instance of Microsoft SQL Server, dedicated to IBM AD components. It is recommended that all IBM AD components to be installed on the same operating system instance where Microsoft SQL Server is running.
- **IBM Application Discovery** can be deployed on Virtual Machines (VMs) as well, if the minimum hardware requirements are met. In a virtual environment, it is highly recommended to have:
  - Microsoft SQL Server and IBM AD Service Components VM Machines located in the same cluster or even better on the same physical hardware. This setup reduces any potential network latency and slowness.
  - A dedicated storage solution (drive) for **IBM Application Discovery** (that includes Microsoft SQL Server VM), instead of a shared solution (with other VMs, products).
- For IBM AD Dedicated Database Server:
  - CPU: Intel XEON Dual processor with minimum of 4 cores each, 3 GHz or higher, with Turbo support.
  - 64 GB of RAM.
  - Dedicated hard disk drive for the operating system, minimum 500 GB. Secondary dedicated hard disk drive for data, minimum 2 TB.

    Note: If the secondary database server is a server with a running instance of Microsoft SQL Server, for an efficient disk space utilization, it is recommended to use 70% of disk space for Database data, and 30% of disk space for log data, such as transaction logs. The two types of data can be stored on the same disk.
- For all the IBM AD components:
  - CPU: Intel XEON Dual processor with minimum of 4 cores each, 3 GHz or higher, with Turbo support.
  - Minimum 32 GB of RAM.
  - Dedicated hard disk drive for the operating system, minimum 500 GB. Secondary dedicated hard disk drive for data, minimum 2 TB.
  - 4 GB allocated Virtual Memory.
- For IBM AD Analyze Client:
  - CPU: Intel i5 or equivalent.
  - Minimum 8 GB of RAM.
  - 20 GB allocated disk space.

© Copyright IBM Corp. 2010, 2022

- For IBM AD Build Client:
  - CPU: Intel i5 or equivalent.
  - 16GB of RAM.
  - 20 GB allocated disk space.

For Evaluation / POC implementations

The minimum recommended hardware requirements for evaluating the IBM AD in a Windows environment:

Note: SQL Server and all the IBM AD components can be installed on this machine for evaluation / POC purposes only.

- For IBM AD Workstation:
  - CPU: Intel Core i5 or equivalent.
  - 32 GB of RAM.
  - 35 GB allocated disk space.

Supported Platforms and Versions

For detailed system, hardware, bit value requirements, and supported software, you can find a software product compatibility report at https://www.ibm.com/software/reports/compatibility/clarity/softwareReqsForProduct.html.

Supported operating systems

IBM AD Analyze Server/Batch Server/Configuration Server can be installed on:

- Windows 10 (64-bit)
- Windows Server 2012/2016/2019
- Linux Red Hat Enterprise (RHEL) Server 7 x86-64/SUSE Enterprise Server (SLES) 15 x86-64/Ubuntu 16.04 LTS x86-64
- Linux Red Hat Enterprise (RHEL) Server 7 IBM z Systems/SUSE Enterprise Server (SLES) 15 IBM z Systems/Ubuntu 16.04 LTS IBM z Systems

Important: Make sure that a Linux® Samba client is set up before installing IBM AD products on Linux.

IBM AD Build/Analyze Client can be installed on:

- Windows 10 (64-bit)
- Windows Server 2012/2016/2019

IBM® AD Connect for Mainframe can be installed on z/OS 2.3/2.4.

Supported databases

The following databases are supported:

- Db2 for z/OS version 12.1 and later
- Db2 Standard Edition 11.5.5.x

  Important: Based on your operating system, it is recommended to download and use one of the following images from the Db2 download page.
  - Windows 64 bit, x86, on AMD64 and INTEL EM64T
  - Linux 64 bit, x86-64 on AMD64 and INTEL EM64T
  - Linux 64 bit, System Z, System z9® or zSeries
- Microsoft SQL Server 2012 and later

Note:

- Only one relational database repository type is taken into account when creating a project. The relational database repository types are:
  - Microsoft SQL Server
  - Db2 for z/OS
  - Db2 for LUW (Linux, UNIX, and Windows)
- Microsoft SQL Server Standard or Enterprise Edition is required for IBM Application Discovery Microsoft SQL Server repository support.
- Microsoft SQL Server Express® Edition can be used for IBM AD Suite evaluation purposes only.
- For more information about the minimum prerequisite level that is required for a specific database and version, see Supported Databases for IBM Application Discovery.

Prerequisite Software and Configurations

- **Eclipse**

  The minimum supported version is 32/64-bit Eclipse for RCP and RAP Developers v4.4.x and it can be downloaded from https://www.eclipse.org/downloads/packages/release/photon/m5/eclipse-rcp-and-rap-developers.

  The latest supported version is 32/64-bit Eclipse IDE for RCP and RAP Developers v4.16.x and it can be downloaded from https://www.eclipse.org/downloads/packages/release/2020-06/r/eclipse-ide-rcp-and-rap-developers.

- **IBM Rational® Development for z Systems® (RDz)**

  The minimum supported version is v9.5.x.

- **IBM Developer for z Systems (IDz)**

  The minimum supported version is v14.2.x.x.

- The supported Internet browsers (used for accessing the **IBM AD Configuration Server**) are: Google Chrome (recommended), Mozilla Firefox, and Microsoft Edge (version 79 or later).

  Note: Make sure to enable JavaScript on your Internet browser of choice.

Java Runtime Environment

Any of the following Java™ Runtime Environments must be installed on your machine:

- 64-bit AdoptOpenJDK 8/11/13 HotSpot - https://adoptopenjdk.net/releases.html

  Important:

  - Make sure to download the **JRE** version of AdoptOpenJDK for the target platform.
  - Make sure that *Set JAVA_HOME variable* and *JavaSoft (Oracle) registry keys* subfeatures are selected when you install AdoptOpenJDK Development Kit with Hotspot.
- 64-bit Java 8/11 Oracle
  - To download 64-bit Java 8 go to https://www.oracle.com/java/technologies/downloads/#java8.
  - To download 64-bit Java 11 go to https://www.oracle.com/java/technologies/downloads/#java11.
- 64-bit Java 8 IBM - https://developer.ibm.com/javasdk/downloads/

  Note:

  - **On Windows**

    Since there is no executable for IBM Java, it needs to be unzipped, and you need to set the IBM Java path in *Windows Registry*.

Important: IBM AD installer is looking in *Windows Registry* to find out the IBM Java path. Use extreme caution when you edit the *Windows Registry*. Incorrect edits can render your computer unusable.

1. Create the following path: **Computer** > **HKEY_LOCAL_MACHINE** > **SOFTWARE** > **JavaSoft** > **Java Development Kit** > **1.8**.
2. Right-click **1.8**, and then select **New** > **String Value**.
3. Rename **New Value #1** to **JavaHome**.
4. Right-click **JavaHome**, and then select **Modify**.
5. In **Value data**, enter the path to the unzipped IBM Java folder.

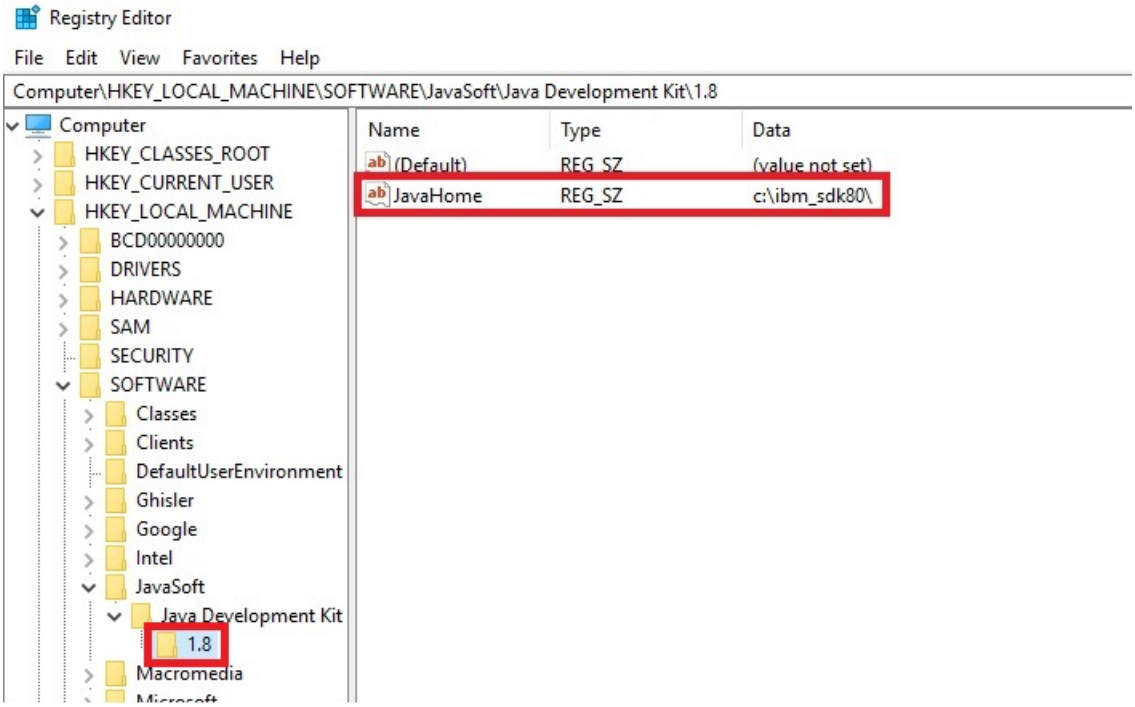


*Figure 2. **IBM Java** path*

Note: ibm_sdk80 represents the IBM Java version that is used during the installation process.

6. Restart the Windows machine so that the changes can take effect.

– **On Linux or Linux for System z®**

Make sure to run the update-alternatives command to set the specified IBM Java location to be used on the system.

Example:

```
update-alternatives --install /usr/bin/java java /opt/ibm/java-s390x-80/bin/java 1
```

Where `/opt/ibm/java-s390x-80/bin/java` represents the location of IBM Java. Based on the IBM Java version that is used during the installation process, the location might change slightly.

Important:

- Make sure to uninstall any previous version of Java before you install the newest version and use the latest service release of Java v1.8. Failing to do this an unexpected IBM AD behavior might result.
- IBM AD Build Client runs only on the 64-bit JVM.

Important: Make sure that the path for Java is set in your system as follows: **Start** > **System** > **Advanced Settings** > **Environment Variables** > **System variables** > **Path**. The path to the Java installation folder must be set to include the `bin` folder as in `C:\Program Files\Java\jre1.8.0_131\bin`.

User Access

User access to workspaces

IBM AD workspaces that are created in IBM AD Configuration Server contain IBM AD projects. When a workspace is created in IBM AD Configuration Server, by default, it is public. Configuring user access to workspaces can restrict user access to projects from IBM AD Analyze Client and IBM AD Build Client. To configure the user access to this workspace, add the credentials of only the users who must have access to its projects. User names are case-sensitive in IBM AD Configuration Server.

Note: IBM AD workspaces are defined within an environment in IBM AD Configuration Server. For any given environment, the workspace set as the default is the workspace where projects are published automatically at build time. Only one IBM AD workspace at a time can be set to be the default for a given environment.

For more information about user access to workspaces, see “Managing workspaces' access rights” on page 489.

Windows service access and User access to folders

The following table shows the required access to folders by different IBM AD server and client components. The access type of write, where specified, implies being able to also delete and create files.

By default on Windows, IBM AD server components, which run as Windows services, are configured to run using the credentials of the Windows Local System account ID of the machine on which they are installed. If any folder locations that are defined in your environment and listed in the following table as requiring access by a server component are not accessible to the IBM AD server's Local System account ID, you can go to the Properties dialog box of one or more of the IBM AD server services, and then configure the services to run using the credentials of a different ID that does have appropriate access to the necessary folder locations. Additionally, when using such a non-default service ID to run IBM AD services, on the server machine, the service ID must be granted the "Log on as a batch job" and "Log on as a service" local rights.

For IBM AD Analyze Client and IBM AD Build Client, grant access to the logged on Windows user that is running the component.

Important: When configuring and using IBM AD components, not only the folder locations, but also the format or nomenclature of how those folder locations are written is important. In most situations, the format of how a folder's location is specified in IBM AD Build Client when creating and updating a project is the same format that is used by other components when trying to access that project folder and the folders and files within it. Similarly, when using z/OS connections and retrieving source assets from a z/OS system by using IBM AD Connect for Mainframe, the format of how the path for the retrieved members is specified in the IBM AD configuration process is used when components access source code and other assets that have been retrieved from z/OS.

Example

An IBM AD server machine has a hostname of addiserver5.companyabc.com, an IP address of 10.1.2.5, and a projects folder. The projects folder is shared so that it can be referenced in your network by a UNC name of either `\\10.1.2.5\Projects` or `\\addiserver5.companyabc.com\Projects`. This shared folder is also mapped by other Windows machines as drive `P:`. When creating and modifying a project using IBM AD Build Client, whichever of those UNC names is used also needs to be configured or used by most IBM AD server components and IBM AD Analyze Clients.

Therefore, if `\\10.1.2.5\Projects` is the format of the shared projects folder path used when creating projects in IBM AD Build Client, to assure that all IBM AD components, both those running on the server and those that might be running remotely such as IBM AD Analyze Client and IBM AD Build Client, can successfully access the IBM AD projects, their sources, and other project and analysis assets, use the same format, which is not `\\addiserver5.companyabc.com\Projects` or `P:`, when configuring IBM AD and creating remote mappings in the configuration information of IBM AD to refer to that folder

location. Similarly, consistently use the same format when configuring source code locations, Index locations, and other IBM AD folder locations.

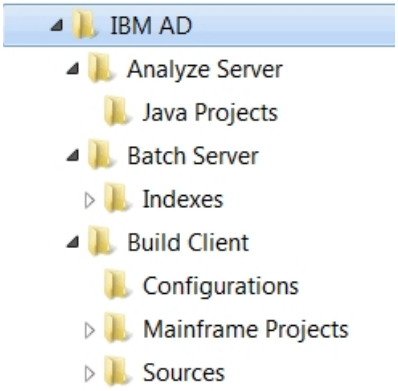
The default projects path value is listed in IBM AD Configuration Server and is used as the default value in IBM AD Build Client when creating a new project. To confirm the default projects path value, browse to the IBM Application Discovery Configuration Servers Admin page at http://localhost:8080, and then click **localhost:2181**>**Install Configurations**>**IBM Application Discovery Build Client**. The value is in the **Default projects path** field.

If a project is created by using a non-default value for its project path, to confirm the path value, browse to the IBM Application Discovery Configuration Servers Admin page at http://localhost:8080, and then click **localhost:2181**>**Environments**>***environment_name***>**Projects**>***project_name***>**Information**. The value is in the **Local project path** field on the **Info** tab.

Component	Folder	Required access	Note
IBM AD Analyze Client	IBM AD Analyze Client user's Eclipse installation folders	Read and write	
	IBM AD Analyze Client user's Eclipse workspace folders	Read and write	
IBM AD Analyze Server (Optional)	IBM AD Analyze Server installation folders	Read and write	The write access is required for the user under which the Windows service or Linux process runs.
	The shared folder for shared Java projects	Read and write	A shared folder for shared Java projects must be created and be referenced in the IBM AD Analyze Server manager.
IBM AD Build Client	Project folders	Read and write	
	Configuration folders for z/OS connections	Read and write	
	Source code folders	Read and write	
IBM AD Batch Server	The `Indexes` folder	Read and write	
	IBM AD Batch Server installation folder	Read and write	
	Project and source folders	Read and write	
IBM AD File Service	The source file folders that are specified through **`local`** parameters in `conf.yaml`	Read	
IBM AD Manual Resolutions Service	The folder of journal files that is specified in `conf.yaml`	Read and write	

Component	Folder	Required access	Note
IBM AD Search Service	The folder of index files that is specified in `conf.yaml`	Read	

The following image shows a best practice example of the IBM AD installation folder structure. However, you can customize the structure based on your particular needs.



Microsoft SQL Server Configurations

Most IBM AD components connect to a Microsoft SQL Server by using SQL Authentication. For Microsoft SQL Server 2012/2014/2016/2017/2019, perform the following configurations:

Important:

- It is highly recommended to enable SSL encryption for an instance of SQL Server by using Microsoft Management Console. For more information, see **How to enable SSL encryption for an instance of SQL Server by using Microsoft Management Console**.
- No specific setting needs to be added for AD when SQL runs in TLS mode, it simply can be used as it is.
- For SQL Server communication to work with TLS 1.2, enable TLS for IBM Java. The IBM JDK/JRE supports TLS 1.2 starting from Java 7 SR 10, and TLS is not enabled by default. To turn it on, add the parameter **`-Dcom.ibm.jsse2.overrideDefaultTLS=true`** on the `eclipse.ini` file.
- For AD components to work with SSL/TLS, the parameter **`-Djsse.enableCBCProtection=false`** needs to be added to the `eclipse.ini` file after the parameter **`-vmargs`**. The same **`enableCBCProtection`** parameter was set in the `startServer.bat` of Batch server 5.1 and later versions. For earlier versions of Batch server, you need to set the parameter.

1. Make sure that Microsoft SQL Server instance is configured with a **case-insensitive (CI)** collation.

   Important: IBM Application Discovery does not support the Microsoft SQL Server instance's collation *Latin1_General_CP1254_CI_AS*.

2. Make sure Microsoft SQL Server Agent service is started.

   a. Setting up an SQL user account:

   Important: The SQL user account must have the privilege to create database, create/delete/update table, create/delete/update index, execute sql clause.

   i) Start SQL Server Management Studio.

   ii) Expand **Security** > **Logins** then right-click **Logins** and choose **New Login**. Add a login name, select SQL Server Authentication, add a password, and make sure that the default database is set to master. This user is referred to as **IBM AD SQL Identity**.

   iii) Expand **Databases** > **System Databases** > **master** > **Security** > **Users** then right-click **Users** and choose **New User**. Choose the **IBM AD SQL Identity** user and click **OK**.

iv) Expand **Databases** > **System Databases** > **master** then right-click **master** and choose **Properties**. Go to **Permissions** tab and for the **IBM AD SQL Identity** and make sure that permissions are granted for: **Create database**, **Create function**, **Create procedure**, **Create table**, and **Create view**.

v) The following permission must be granted only if the **Rename project** feature is used in IBM AD Build, otherwise it is not needed. Right-click on the **SQL server** instance and then select **Properties**. In the **Server Properties** window select **Permissions**: From the roles list, select the **IBM AD SQL Identity** and then select **Grant** for *Alter any database* permission.

3. Configuring SQL Server to enable it to accept connection over TCP/IP:

   a. Start SQL Server Configuration Manager.

   b. Select **SQL Native Client Configuration (32bit)** > **Client Protocols** and then right-click **TCP/IP** and set it to **Enabled**.

   c. Select **SQL Server Network Configuration** > **Protocols for <Instance ID>** and make sure that **Shared Memory** and **TCP/IP** are set to **Enabled**.

   d. Select **SQL Native Client Configuration** > **Client Protocols** and make sure that **Shared Memory** and **TCP/IP** are set to **Enabled**.

   e. Close SQL Server Configuration Manager.

   Important: SQL Server Configuration Manager writes startup parameters to the registry. They take effect upon the next startup of the SQL Server.

IBM Db2 for z/OS Server Configurations

On workstations where IBM AD Build Client is installed, Data Server Client V11.1 that contains IBM Data Server Provider for .NET must be installed and licensed. The mentioned package is part of Db2 Connect™ V11.1 product.

Note: .NET framework 4.5.1 or higher is prerequisite for IBM AD Build Client.

To license the .NET provider, click **Start**, select **Run**, type **cmd** to open the command window and run the following command:

```
"<Installation Path for IBM Data Server Client>\BIN\db2licm.exe" -a <path to license
file>\<name of license file>
```

Important: The name of the license file is given by the Db2 Connect™ license type. The following Db2 Connect™ products are available, together with the corresponding license's name:

- **DB2® Connect Personal Edition** - `db2conpe.lic`
- **DB2 Connect Application Server Edition** - `db2consv_as.lic`
- **DB2 Connect Enterprise Edition** - `db2consv_ee.lic`
- **DB2 Connect Unlimited Edition for System i®** - `db2consv_is.lic`
- **DB2 Connect Unlimited Edition for System z** - `db2consv_zs.lic`

Example when **DB2 Connect Enterprise Edition** is used:

```
"C:\Program Files\IBM\SQLLIB\BIN\db2licm.exe" -a
C:\Apps\DB2_CEE_QS_Activation_11.1\consv_ee\db2\license\db2consv_ee.lic
```

In case that IBM AD Build Client is intended to be used by a standard user, with no administration privileges, make sure that during Db2 Connect V11.1 installation process the **Enable operating system security** check box is selected.

Note: If you selected the **Enable operating system security** check box, you now have to add users to the *DB2ADMNS* or the *DB2USERS* groups for users that need to run IBM AD Build Client. For more information, see Adding your user ID to the DB2ADMNS and DB2USERS user groups.

Some packages must be bound on the server. The bound files are listed in the `ddcsmvs.lst` file for IBM Z, for example, `C:\Program Files\IBM\SQLLIB_01\bnd\ddcsmvs.lst`.

User permissions

In IBM AD application, a z/OS user account (Authentication ID) is used to work with the Db2 for z/OS repository. The user account must have the privilege to execute sql clauses and must have permissions to perform the following actions:

- Use the storage group that is specified when configuring the Db2 z/OS connection in **IBM AD Configuration Server**.

  Note: Make sure that the storage group is initially created.
- Use the SYSDEFLT default storage group.
- Use the default buffer pools.
- Create and use databases, tables, table spaces, indexes, constraints, views, triggers, stored procedures, and user-defined functions.
- Run the **SELECT** command over system tables (`SYSIBM.SYS*`).
- Have full control over the databases that are created by the user account.
- `GRANT BINDADD, BINDAGENT, CREATEDBA, CREATESG, DISPLAY`
- `GRANT ALTERIN, CREATEIN, DROPIN ON SCHEMA *`
- `GRANT CREATE IN COLLECTION *`
- `GRANT SELECT ON TABLE SYSIBM.SYSROUTINES_SRC`
- `GRANT SELECT ON TABLE SYSIBM.SYSROUTINES_OPTS`

Important: All permissions must be granted directly to the user account (Authentication ID).

Db2 for LUW (Linux, UNIX, and Windows) prerequisites

This topic lists the prerequisite software for creating the Db2 for LUW databases.

IBM Db2 for LUW Server Configurations

On workstations where IBM AD Build Client is installed, Data Server Client V11.5 that contains IBM Data Server Provider for .NET must be installed and licensed. The mentioned package is part of Db2 Connect™ V11.5 product.

Note: .NET framework 4.5.1 or higher is prerequisite for IBM AD Build Client.

To license the .NET provider, click **Start**, select **Run**, type **cmd** to open the command window and run the following command:

```
"<Installation Path for IBM Data Server Client>\BIN\db2licm.exe" -a <path to license
file>\<name of license file>
```

Important: The name of the license file is given by the Db2 Connect™ license type. The following Db2 Connect™ products are available, together with the corresponding license's name:

- **DB2 Connect Personal Edition** - `db2conpe.lic`
- **DB2 Connect Application Server Edition** - `db2consv_as.lic`
- **DB2 Connect Enterprise Edition** - `db2consv_ee.lic`
- **DB2 Connect Unlimited Edition for System i** - `db2consv_is.lic`
- **DB2 Connect Unlimited Edition for System z** - `db2consv_zs.lic`

Example when **DB2 Connect Enterprise Edition** is used:

```
"C:\Program Files\IBM\SQLLIB\BIN\db2licm.exe" -a
C:\Apps\DB2_CEE_QS_Activation_11.1\consv_ee\db2\license\db2consv_ee.lic
```

In case that IBM AD Build Client is intended to be used by a standard user, with no administration privileges, make sure that during Db2 Connect V11.5 installation process the **Enable operating system security** check box is selected.

Note: If you selected the **Enable operating system security** check box, you now have to add users to the *DB2ADMNS* or the *DB2USERS* groups for users that need to run IBM AD Build Client. For more information, see Adding your user ID to the DB2ADMNS and DB2USERS user groups.

Some packages must be bound on the server. The bound files are listed in the `ddcsmvs.lst` file for IBM Z, for example, `C:\Program Files\IBM\SQLLIB_01\bnd\ddcsmvs.lst`.

For more information, see Registering a license certificate file using the db2licm command.

Prerequisite software for automatically creating the Db2 for LUW databases

For Cross Applications database, the following prerequisites need to be installed on the machine where IBM AD Servers and Services will be installed:

- IBM Data Server Client V11.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (for Windows only)
- Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 (for Windows only)

For Annotations database, the following prerequisites need to be installed on the machine where IBM AD Servers and Services will be installed:

- IBM Data Server Client V11.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (for Windows only)
- Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 (for Windows only)

User permissions

In IBM AD application, an operating system user account (Authentication ID) is used to work with the Db2 for LUW repository. The user account must have **SYSCTRL** authority in order to be able to create databases and to perform actions on the created databases. For more information, see the System control authority (SYSCTRL) topic.

IBM AD Connect for Mainframe Prerequisites

IBM AD Connect for Mainframe can be installed on:

- z/OS version 2.3 or later.
- The maximum storage space is 5 cylinders.

Before installing IBM AD Connect for Mainframe on the host machine (mainframe), the following authorizations are required. IBM AD Connect for Mainframe does not update any mainframe resources. All the following authorizations are for READ access only.

1. Authorization to add the load library of IBM AD Connect for Mainframe to APF.
2. Authorization for running the listener of IBM AD Connect for Mainframe.
3. Authorization to access all libraries specified in the STEPLIB card (see “Configuring the Listener PROC” on page 152).
4. Authorization to access TCP connections to IBM AD Connect for Mainframe. As part of this requirement, the RACF® user ID assigned to the IBM AD Connect for Mainframe started task must contain an OMVS segment.
5. Additional required authorizations according to analyzed area:

Analyzed Area	Required Authorization
Adabas	Authorization to issue an ADAREP command.

Analyzed Area	Required Authorization
Control-M	Access to the libraries that contains the control M data.
DB2	Rights to read from the Db2 system tables (SYSIBM).
SMF	Access to the SMF dump files.
Libraries and Members	Access to the libraries.
Natural	Authorization to issue a Natural batch command and read Access to all Natural libraries (LOGON).
Operator commands	Normal RACF security to allow the user to issue those commands.
WebSphere® MQ	Authorization to perform PUT and GET from command and reply queues.
PDS libraries	Read-only access to the source libraries (for members not stored in Endevor)
CA Endevor	Authorization to use the CA Endevor API, used only for reading the list of members and retrieving a copy of those members. Authorization to access CA Endevor libraries and control files that are used during API processing.
CA-7	Read-only access the CA-7 initialization parameters member, Authorization to execute the commands: `LJOB,JOB=*,LIST=ALL` – lists all Jobs information, `LGVAR,JOB=*,LIST=ALL` – lists all global variables for Jobs.

IBM AD Web Services Prerequisites

Before you install and configure IBM AD Catalog, IBM AD Audit and IBM AD BRD web services, make sure that the following steps are performed:

- Any of the following Java Runtime Environments is installed:
  - 64-bit AdoptOpenJDK 8/11/13 HotSpot
  - 64-bit Java 8/11 Oracle
  - 64-bit Java 8 IBM

  For more information, see “Java Runtime Environment ” on page 9.
- Any of the following database servers is installed and configured:
  - **Microsoft SQL Server**

    Microsoft SQL Server 2012/2014/2016/2017 is installed and configured (accessible, if it is installed on a different machine). For more information, see “Microsoft SQL Server Configurations” on page 13 at the end of this topic.
  - **Db2 for LUW (Linux, UNIX, and Windows)**

    If you use **Db2 for LUW (Linux, UNIX, and Windows)** you must **manually** create the Audit database before you run the IBM AD Configuration Wizard. To create the Audit database you can use the Db2 database client or you can start the Db2 command window and execute the following clause:

    ```
    db2 create database EZAUDIT using codeset UTF-8 territory en PAGESIZE 16384
    ```
  - **Db2 for z/OS**

    If you use **Db2 for z/OS**, you must **manually** create the Audit database before you run the IBM AD Configuration Wizard or define a relational database server against Db2 for z/OS. To create the Audit database, complete the following steps:

1. Run the following SQL clauses to create the Audit database, tables, and indexes.

```
--#SET TERMINATOR ;

CREATE DATABASE EZAUDIT
--STOGROUP CUSTOM_STOGROUP_NAME
CCSID UNICODE ;

SET CURRENT SCHEMA = 'EZAUDIT';
CREATE TABLE APPLICATIONS (
     ID          INTEGER NOT NULL,
     DESCRIPTION VARCHAR(255),
     NAME        VARCHAR(255),
     CONSTRAINT APPLICATIONS_PK PRIMARY KEY(ID)
) IN DATABASE EZAUDIT;
CREATE TABLE EVENTS (
     ID                INTEGER GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY
1),
     APPLICATION_ID    INTEGER,
     DATE              TIMESTAMP,
     INSTANCE_ID       VARCHAR(255),
     IP                VARCHAR(255),
     OPERATION_NAME    VARCHAR(255),
     OPERATION_TYPE_ID INTEGER,
     PROJECT_NAME      VARCHAR(255),
     USER_NAME         VARCHAR(255),
     CONSTRAINT EVENTS_PK PRIMARY KEY(ID)
) IN DATABASE EZAUDIT;
CREATE TABLE EVENTS_ARCHIVED (
     ID                INTEGER GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY
1),
     APPLICATION_ID    INTEGER,
     DATE              TIMESTAMP,
     INSTANCE_ID       VARCHAR(255),
     IP                VARCHAR(255),
     OPERATION_NAME    VARCHAR(255),
     OPERATION_TYPE_ID INTEGER,
     PROJECT_NAME      VARCHAR(255),
     USER_NAME         VARCHAR(255),
     CONSTRAINT EVENTS_ARCHIVED_PK PRIMARY KEY(ID)
) IN DATABASE EZAUDIT;
CREATE TABLE EVENTS_INFO (
     EVENT_ID   INTEGER NOT NULL,
     INFO_VALUE VARCHAR(255),
     INFO_KEY   VARCHAR(255) NOT NULL,
     CONSTRAINT EVENTS_INFO_PK PRIMARY KEY(EVENT_ID, INFO_KEY)
) IN DATABASE EZAUDIT;
CREATE TABLE EVENTS_INFO_ARCHIVED (
     EVENT_ARCHIVED_ID INTEGER NOT NULL,
     INFO_VALUE        VARCHAR(255),
     INFO_KEY          VARCHAR(255) NOT NULL,
     CONSTRAINT EVENTS_INFO_ARCHIVED_PK PRIMARY KEY(EVENT_ARCHIVED_ID, INFO_KEY)
) IN DATABASE EZAUDIT;
CREATE TABLE NOTIFICATIONS (
     ID             BIGINT GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY 1),
     CREATED        TIMESTAMP NOT NULL,
     MESSAGE_KEY    VARCHAR(255),
     MESSAGE_VALUES VARCHAR(255),
     SOURCE         VARCHAR(36) NOT NULL,
     TITLE          VARCHAR(255) NOT NULL,
     TYPE           INTEGER,
     CONSTRAINT NOTIFICATIONS_PK PRIMARY KEY(ID)
) IN DATABASE EZAUDIT;
CREATE TABLE USAGE_DETAIL (
     ID         BIGINT GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY 1),
     END_TIME   TIMESTAMP,
     LOG_TIME   TIMESTAMP,
     START_TIME TIMESTAMP,
     SUBTYPE    VARCHAR(255),
     TYPE       VARCHAR(255),
     VALUE      BIGINT,
     CONSTRAINT USAGE_DETAIL_PK PRIMARY KEY(ID)
) IN DATABASE EZAUDIT;
CREATE TABLE USER_PROFILE (
     ID        BIGINT GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY 1),
     LAST_SEEN TIMESTAMP NOT NULL,
     USERNAME  VARCHAR(255) NOT NULL,
     CONSTRAINT USER_PROFILE_PK PRIMARY KEY(ID),
     CONSTRAINT USER_PROFILE_USERNAME_UNQ UNIQUE (USERNAME)
) IN DATABASE EZAUDIT;
CREATE TABLE USERS (
```

```
    ID        BIGINT GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY 1),
    CREATED   TIMESTAMP NOT NULL,
    LICENSE_TYPE    INTEGER,
    USERNAME  VARCHAR(255) NOT NULL,
    CONSTRAINT USERS_PK PRIMARY KEY(ID),
    CONSTRAINT USERS_USERNAME_UNQ UNIQUE (USERNAME)
) IN DATABASE EZAUDIT;
CREATE INDEX IN_SOURCE_CREATED ON NOTIFICATIONS (CREATED, SOURCE, ID);
ALTER TABLE EVENTS_INFO ADD CONSTRAINT EVENTS_INFO_EVENTS_FK FOREIGN KEY (EVENT_ID)
REFERENCES EVENTS
ENFORCED;
ALTER TABLE EVENTS_INFO_ARCHIVED ADD CONSTRAINT EVENTS_INFO_ARCHIVED_EVENTS_ARCHIVED_FK
FOREIGN KEY (EVENT_ARCHIVED_ID)
REFERENCES EVENTS_ARCHIVED
ENFORCED;
```

2. Copy the `db2jcc_license_cisuz.jar` file from your Db2 into the *<IBM ADDI Installation Folder>*`/IBM AD Web Services/wlp/usr/shared/config/lib/global` directory.

- **9080** and **9443** are the default ports that are used by the **IBM Application Discovery WebSphere Liberty Profile Service**. Make sure that both ports are open, unused, and not blocked by your firewall. If the default ports are not available, you can check the troubleshooting tips at the end of this topic.
- Starting with IBM AD V5.1.0.7, you don't need to perform the pre-configurations of IBM WAS Liberty Web Server and the pre-configurations of SQL database for IBM AD Audit. Audit database is created **automatically** for Microsoft SQL Server users.
- If you use **Db2 for LUW (Linux, UNIX, and Windows)** you must **manually** create the Catalog database. To create the Catalog database you can use the Db2 database client or you can start the Db2 command window and execute the following clause:

  ```
  db2 create database CATDB using codeset UTF-8 territory en PAGESIZE 16384
  ```

  Note: Once the Catalog database has been created make sure to configure the Catalog Service to run on Db2 for LUW with TLS. For more information see, Configuring the Catalog Service to run on Db2 for LUW with TLS.
- An SQL database needs to be manually created by your database administrator for IBM AD Catalog.

  Note:

  – The name of this database must match the name of the database that will be added in the `server.xml` configuration file, during the configuration process of IBM AD Catalog. For more information, see “Configuring the Catalog Service” on page 58.

  – The database user that is used in the `server.xml` configuration file and connects to the Catalog database needs to have the *db_owner* database role membership.

  How to set up manually the Catalog database

  1. Start SQL Server Management Studio.
  2. Log in by using the username and password for the **IBM AD SQL Identity**, as defined in “Microsoft SQL Server Configurations” on page 13.
  3. Right-click **Databases** and choose **New Database**.
  4. Enter the **Catalog** database name, for example: **catalog_db** and press **OK**.
  5. As a result the **catalog_db** database is created.

     Important: The Catalog database will be automatically populated when **Data Collector** starts for the first time.

For **Data collector** the version of the target **z/OS Connect** server must be 3.0.2 or higher.

Troubleshooting tips: Changing the default ports for IBM Application Discovery WebSphere Liberty Profile Service

IBM Application Discovery WebSphere Liberty Profile Service uses **9080** and **9443** as the default ports. If the default ports are not available, then determine the ports that are available and that you

would like to assign to **IBM Application Discovery WebSphere Liberty Profile Service**. When the ports are determined, complete the following steps to update the default ports for **IBM Application Discovery WebSphere Liberty Profile Service**.

1. Install IBM ADDI as described in “ Installing components with the IBM ADDI Installer” on page 25. By default, for clean installations, the installer will open the **IBM AD Configuration Wizard** in the system's default browser. But since the default ports for **IBM Application Discovery WebSphere Liberty Profile Service** are not available, it wouldn't start correctly and the **IBM AD Configuration Wizard** would report an error similar to "This page cannot be displayed" or "Unable to connect" or "This site can’t be reached" depending on system's default browser.
2. To confirm if it's a port conflict issue, open the `console.log` file under the *<IBM ADDI Installation Folder>*`/IBM AD Web Services/wlp/usr/servers/ad_server/log/` directory, and check if you notice error messages that are similar to the following example:

   ```
   [ERROR   ] CWWKO0221E: TCP Channel defaultHttpEndpoint initialization did not succeed.  The
   socket bind did not succeed for host * and port 9080.  The port might already be in use.
   Exception Message: Address already in use: bind
   [ERROR   ] CWWKO0221E: TCP Channel defaultHttpEndpoint-ssl initialization did not succeed.
   The socket bind did not succeed for host * and port 9443.  The port might already be in
   use.  Exception Message: Address already in use: bind
   ```

3. When the port conflict issue is confirmed, stop the **IBM Application Discovery WebSphere Liberty Profile Service**. The service can be stopped from Windows Services (services.msc) by locating **IBM Application Discovery WebSphere Liberty Profile Service** and clicking **Stop**.
4. Open the `server.xml` file and update the `httpPort` and `httpsPort` properties in the following section.

   ```
   <httpEndpoint httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint" host="*" />
   ```

   Example:

   ```
   <httpEndpoint httpPort="<new http port>" httpsPort="<new https port>"
   id="defaultHttpEndpoint" host="*" />
   ```

   The *<new http port>* and *<new https port>* in the example represents the new port numbers that you want to use.
5. Save the `server.xml` file with your changes.
6. Restart the **IBM Application Discovery WebSphere Liberty Profile Service**. The service can be started from Windows Services (services.msc) by locating **IBM Application Discovery WebSphere Liberty Profile Service** and clicking **Start**.
7. Wait for the **IBM Application Discovery WebSphere Liberty Profile Service** and Audit application to start successfully.
8. Open the browser and visit the **IBM AD Configuration Wizard** at `http://`*<host or ip address>*`:<new http port>/ad-audit/setup`. For example, if you want to use `localhost` as the host address and **9090** as the `httpPort`, then you can access the **IBM AD Configuration Wizard** at `http://localhost:9090/ad-audit/setup`.

TCP Port Requirements and Firewall Exceptions

The following table summarizes the TCP ports that need to be allowed by the firewall in order for the Application Discovery Suite to function as intended.

In all cases, communication is bidirectional. The firewall must allow both the incoming traffic, which represents requests, for the mentioned ports, and the outgoing traffic, which represents the answers to these requests.

From (Sender)	To (Listener Component)	Default Listener Port	Note
• AD Analyze Clients • AD Audit Service • AD Batch Server • AD Build Client • AD Build Configuration • AD Catalog	SQL Server	TCP 1433	The port of the SQL Server Database Engine instance that hosts the AD databases. Majority of the AD components use this port to read/write data from/into the SQL databases. The default instance of the SQL Server Database Engine listens on TCP port 1433, but it can be changed via SQL Server admin tools. Ask your database server administrator what port is used by the SQL server instance that is used by AD. Make sure not to use TCP port 1434, which is used by Dedicated Administration Console (DAC).
The computer where the browser session is opened	AD Configuration Server	TCP 8080	The port that is used to access the web interface of AD Configuration Server. The default port is 8080, but it can be changed in the `admin-ws.properties` file, which is located in the `conf` folder where AD Configuration Server is installed. If the web interface is accessed only locally on AD Configuration Server, this port does not have to be opened in the firewall.
• AD Analyze Clients • AD Batch Server • AD Build Client • AD Build Configuration • AD Validation Server	AD Configuration Server	• TCP 2181 • TCP 2281 (*)	The port that AD Configuration Server listens on for requests from various AD components that need to obtain the configuration settings from AD Configuration Server. The default port is 2181, but it can be changed in the `server.properties` file, which is located in the `conf` folder where AD Configuration Server is installed. (*) If the `ssl` implementation is used, the default port is 2281 and you need to secure Apache ZooKeeper SSL connections. For more information, see “Securing Apache ZooKeeper SSL connections” on page 203.
AD Analyze Clients	AD Batch Server	• TCP 2424 - TCP 2430 • TCP 2434 - TCP 2440 (*)	The port of the OrientDB database instance that is hosted by AD Batch Server. AD Analyze Client makes requests to this port for retrieving the data that is related to callgraph analyses. OrientDB uses the first free TCP port in the range 2424 - 2430. This can be changed in the `config/orientdb-server-config.xml` file. (*) If the `ssl` implementation is used, the default port for OrientDB SSL is 2434. You need to change your port range to `2434-2440` in the `config/orientdb-server-config.xml` file. For more information, see Configuring OrientDB for SSL/TLS.

From (Sender)	To (Listener Component)	Default Listener Port	Note
AD Analyze Clients	AD Analyze Server	TCP 1099	The port where the Remote Method Invocation (RMI) registry can be found on AD Analyze Server. AD Analyze Clients make RMI-specific requests to this port. The default port is 1099, but it can be changed from AD Analyze Server Manager. To change the setting, click **Server Settings > RMI Registry Port**. AD Analyze Server is required only when AD Analyze Clients use and run Java-specific analyses, so if AD Analyze Server is not installed, the port does not have to be opened in the firewall.
AD Analyze Clients	AD Analyze Server	TCP 1900	The port that is used by AD Analyze Server for the Remote Method Invocation (RMI) communication. AD Analyze Clients make requests to this port for obtaining various information that is needed for Java analyses. The default port is 1900, but it can be changed from AD Analyze Server Manager. To change the setting, click **Server Settings > Export Port**. AD Analyze Server is required only when AD Analyze Clients use and run Java-specific analyses, so if AD Analyze Server is not installed, the port does not have to be opened in the firewall.
AD Analyze Server	AD Analyze Client	A random free TCP port, which is allocated at run time, in the ephemeral port range 49152 - 65535.	The port that AD Analyze Client listens on for notifications from AD Analyze Server. This port is temporarily used, which is released after the application is closed. The default port is a random free TCP port, which is allocated at run time, in the ephemeral port range 49152 - 65535, but AD Analyze Client can be configured to use a specific port. To configure the setting, click **Window > Preferences > Application Discovery > Local Settings > General Settings > Client Settings** in Eclipse, and then set a value in the range 1 - 65535 in the **Port** field. Value 0 means that a free port on AD Analyze Client is selected at run time in the ephemeral port range 49152 - 65535. AD Analyze Server is required only when AD Analyze Clients use and run Java-specific analyses, so if AD Analyze Server is not installed, the port does not have to be opened in the firewall.
• AD Build Client	AD Connect for Mainframe	Any available	The port that AD Connect for Mainframe listens on. It is used by AD Build Configuration to

From (Sender)	To (Listener Component)	Default Listener Port	Note
• AD Build Configuration		TCP port (no default value)	retrieve source code information and operational information from the mainframe, and used by AD Build Client to retrieve source code files from the mainframe. For how to set or change the port that is used by AD Connect for Mainframe, see section "Configuring the Listener PROC" on page 152. There is no default port that is specified. Any available port can be selected. For example, port 6000 or port 46000. After you change this port in AD Connect for Mainframe, the z/OS connection setup needs to be reconfigured to use the new port. To configure the setting, click the **zOS** tab in the AD Build Configuration tool.
AD Connect for Mainframe	AD Validation Service	Any available TCP port (no default value)	The port that AD Validation Service listens on for validation requests from AD Connect for Mainframe. It can be configured in the `ServicePort.txt` configuration file that is located in the AD Validation Server installation folder. No default port is set by default. Any available TCP port can be used. For example, port 48000. AD Validation Service is an optional component. If it is not used, this port does not have to be opened in the firewall.
• AD Analyze Clients • AD Batch Server • AD Build Client	AD Audit Service	TCP 9080	The port that AD Audit Service listens on to receive requests from various AD components for logging audit events. The port number can be changed by altering the httpPort value in the `server.xml` file. The file is located in the folder of the IBM Liberty instance that hosts AD Audit Service. After you change this port, make sure to reconfigure the AD components that audit events to use the new port. For more information, see *IBM AD Web Services User Guide*. The AD Audit and AD Catalog services are optional AD components. They are both hosted by the same WebSphere Liberty instance. If neither of them is used, the port does not have to be opened in the firewall.
• AD Analyze Clients • AD Data Collector	AD Catalog Service	TCP 9080	The port that AD Catalog Service listens on. This port is used by AD Data Collector to push data into AD Catalog, and it is used by AD Analyze Clients to retrieve the data that is needed for displaying API analyses.

From (Sender)	To (Listener Component)	Default Listener Port	Note
			The port number can be changed by altering the httpPort value in the `server.xml` file. The file is located in the folder of the IBM Liberty instance that hosts AD Catalog Service. After you change this port, make sure to reconfigure AD Data Collector and AD Analyze Client to use the new port. For more information, see *IBM AD Web Services User Guide*. The AD Audit and AD Catalog services are optional AD components. They are both hosted by the same WebSphere Liberty instance. If neither of them is used, the port does not have to be opened in the firewall.
• AD File Service • AD Analyze Client • AD Search Service • AD Manual Resolutions Service • AD Mainframe Projects Service • AD Cross Applications Service • Secure Storage	Authentication Server (DEX)	TCP 7600	The default port on which **Authentication Server (DEX)** listens to different requests is 7600. It can be modified in the `conf.yaml` file.
• (Optional) Authentication Server (DEX) • AD Analyze Client	AD File Service	TCP 7700	The default port on which **AD File Service** listens to different requests is 7700. It can be modified in the `conf.yaml` file.
• (Optional) Authentication Server (DEX) • AD Analyze Client	AD Search Service	TCP 7800	The default port on which **AD Search Service** listens to different requests is 7800. It can be modified in the `conf.yaml` file.
• (Optional) Authentication Server (DEX) • AD Analyze Client • AD Batch Server	AD Manual Resolutions Service	TCP 7900	The default port on which **AD Manual Resolutions Service** listens to different requests is 7900. It can be modified in the `conf.yaml` file.
• (Optional) Authentication Server (DEX) • AD Analyze Client • AD Batch Server	AD Mainframe Projects Service	TCP 7650	The default port on which **AD Mainframe Projects Service** listens to different requests is 7650. It can be modified in the `conf.yaml` file.

From (Sender)	To (Listener Component)	Default Listener Port	Note
• (Optional) Authentication Server (DEX) • AD Analyze Client • AD Batch Server	AD Cross Applications Service	TCP 7850	The default port on which **AD Cross Applications Service** listens to different requests is 7850. It can be modified in the `conf.yaml` file.
Authentication Server (DEX)	AD Analyze Client	TCP 9999	The port that is used by **Authentication Server (DEX)**, opened on all Analyze Client machines (in environments using DEX), and used for callback.

Note: Make sure that the firewall does not prevent AD Analyze Client from communicating with AD Batch Server, AD Configuration Server, AD Analyze Server, and the relational database server. Program rules in the firewall might need to be created to allow both the inbound and outbound traffic for the `eclipse.exe` instance on each AD Analyze Client that is located under the installation folder of your Eclipse or IDz instance.

Installing components with the IBM ADDI Installer

To install IBM AD on Windows or Linux, use the IBM ADDI installer. You can also use the IBM ADDI installer to install IBM ADDI Extensions.

Procedure

1. To run the IBM ADDI installer, double-click the `IBM_Application_Discovery_and_Delivery_Intelligence_Installer-6.1.0.0.exe` file.
2. On the **Welcome** page, click **Next**.
3. On the **Licensing Agreements** page, click **I accept the terms of this license agreement**, and then click **Next**.
4. On the **Installation Path** page, specify the installation path, and then click **Next**. The default installation path is `C:\Program Files\IBM Application Discovery and Delivery Intelligence`.

   If the installation path that you specify does not exist, the target directory is created. Confirm the path, and click **OK** in the **Message** dialog box.
5. On the **Select Installation Components** page, select the components that you want to install, and then click **Next**. The components that are not applicable for the current system cannot be selected.

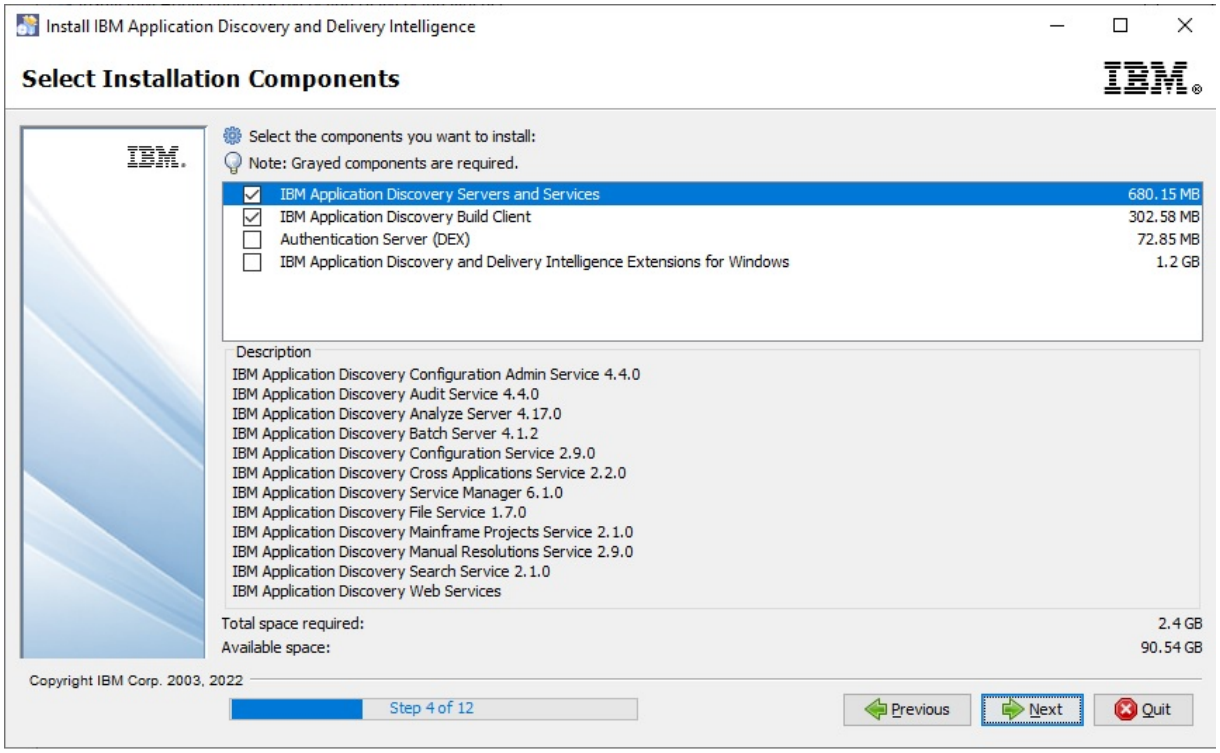


*Figure 3. **Select Installation Components** wizard page*

6. If the IBM Application Discovery and Delivery Intelligence Extensions (IBM ADDI Extensions) check box was selected in step 5, the **Information** page is displayed. Read the information, then click **Next**.
7. If the IBM Application Discovery and Delivery Intelligence Extensions (IBM ADDI Extensions) check box was selected in step 5, the **IBM Application Discovery and Delivery Intelligence Extensions Installations Path** page is displayed. Specify the IBM ADDI Extensions installation path, and then click **Next**.

   If the installation path that you specify does not exist, the target directory is created. Confirm the path, and click **OK** in the **Message** dialog box.

   Note: The default installation path is `C:\Program Files\IBM Application Discovery and Delivery Intelligence\Application Discovery and Delivery Intelligence Extensions\adi6000`.
8. On the **User Data** pages, perform the following actions, then click **Next**.
   a. On the **IBM Application Discovery Configuration Wizard** page, make sure to select the **IBM Application Discovery Configuration Wizard** check box. The IBM AD Configuration Wizard will open in the system's default browser after the installation is completed.

      Note: Depending on the machine performance, you might see the **Context root not found** page in your browser. Wait for a minute and then refresh the page until you see **IBM AD Configuration Wizard** showing in your browser.
   b. Specify the configuration service IP address and service port for IBM AD Build Client.
9. On the **Setup Shortcuts** page, select the shortcuts that you want to create, and then click **Next**.
10. Additionally, after the installation is completed choose the **Generate an automatic installation script** option to create an installation script, where the installation parameters are saved in a `*.xml` file, that can be used later for silent installations. A **Save** dialog box is displayed, allowing to choose the location and name of the installation script. By default, the name of the installation script is `auto-install.xml`.

Important: Once the installation and configuration of IBM Application Discovery and Delivery Intelligence is completed, make sure to activate the full functionality of **IBM AD** by using the latest Activation Kit

available on **Passport Advantage®**. For more information, see Activating the full functionality of IBM AD from a free version.

Installing components by using CLI

In case you do not have access to a graphic interface, follow this procedure to install IBM ADDI:

1. Navigate to the **IBM ADDI** installation path and open a command line.
2. For regular installation, run the following command:

```
java –jar "<installer name>"
```

3. For a silent installation, run the following command:

```
java –jar "<installer name>" -f "<path to install xml file>"
```

Note:

- The `<installer name>` represents the full name including the `.exe` extension.
- To install in silent mode, make sure that an interactive installation was initially performed and that the automatic installation script was generated. For more information, see step 8 in “ Installing components with the IBM ADDI Installer” on page 25.

Installing IBM AD Analyze Client

Installation procedure



Attention:

- Only one instance of **AD Analyze Client** can be used on the same machine in case that **Authentication Server (DEX)** is used for the authentication and authorization.
- If you want to connect the **AD Analyze Client** software on a machine or operating system instance to multiple AD Server instances that are all running the **same** AD Server code level, one copy of the **AD Analyze Client** software can be installed into an `Eclipse/IDz` instance, but each AD server connection (IP/Hostname, port, and Unique ID, at a minimum) must be configured in a unique workspace that does not already contain an instance of the **AD Analyze Client** software's configuration settings.
- If you want to install multiple code levels of the **AD Analyze Client** software on the same machine or operating system instance (to connect to different AD Servers running **different** AD code levels), each level of the **AD Analyze Client** software can be installed into a separate `Eclipse/IDz` instance, and also must use a unique workspace that does not already contain the **AD Analyze Client** software's configuration settings. A workspace in use when the **AD Analyze Client** software is installed and configured contains the **AD Analyze Client** software's configuration settings in a folder named: `<workspace location>\.metadata\.ez\.settings`.

1. To install **IBM AD Analyze Client**: In your Eclipse instance, select **Window** > **Preferences**.
2. From the Install /Update section to the left of the **Preferences** dialog window select **Available Software Sites**: A list of software sites available for update or install is displayed.
3. To select the location from where **IBM AD Analyze Client** is being installed click **Add**: The Add Site dialog window is displayed.
   - In the Name field, enter a name for your **IBM AD Analyze Client** installation.
   - If you extracted the installation archive that you received from IBM and you stored it on your computer, use **Local** and point to the Repository folder generated after the extract operation.
   - If you did not extract the installation archive received from IBM, use **Archive** button to select the installation archive stored on your computer.

- If you did not store the installation archive locally but in a location on your intranet, enter the full path to that location in the **Location** field.

Click **OK**: **IBM AD Analyze Client** will be added to the list of Available Software Sites.

Click **OK** to close the **Preferences** dialog window and proceed to the next step in the installation process.

4. In your Eclipse client select **Help** > **Install New Software**: The Install dialog window is displayed. In **Work with** field select the **IBM AD Analyze Client** site you have defined in the previous step. After you selected the **IBM AD Analyze Client** site, the corresponding **IBM AD Analyze** components are displayed in the central part of the dialog window. By default, all the components are selected.
   a. If you are installing **AD Analyze Client** into IBM IDz, you can choose all the features listed under **IBM AD Analyze**. For more information, see "Explanations of each of the Analyze Client features" on page 28.
   b. If you are installing **AD Analyze Client** into an Eclipse package that is not IBM IDz, for example an Eclipse distribution downloaded from eclipse.org, you can choose the features listed under **IBM AD Analyze**, except you should deselect all features that start with *Application Discovery Integration with* to avoid errors during the installation process. For more information, see "Explanations of each of the Analyze Client features" on page 28.
   c. Once you have selected the correct features to install in your environment, click **Next**.
5. The **Install details** dialog window is displayed.

   Select a component from the list to display a detailed description of it in the **Details** section of the dialog window. Click **Next**
6. The **Review license** dialog window is displayed. Carefully read the License agreement then select **I accept the terms** and press **Finish** to start the installation process.
7. After the installation is completed, Eclipse will prompt you for a restart: Accept the restart operation to see the newly installed features.

   Important: After **IBM AD Analyze Client** has been installed, make sure to configure properly your **Eclipse** or **IDz** instance. For more information, see "Configuring IBM AD Analyze Client" on page 149.

Explanations of each of the Analyze Client features

Note: The translation support features that are named similarly to some of the integration features enable the translation and display of labels in IBM AD Analyze Client into non-English languages that IBM AD supports. If you anticipate that you will use a non-English language in your Analyze Client instance, those translation support features that correlate with any other features will be useful.

- **Application Discovery mainframe analysis** - This feature provides the core mainframe-centric (as opposed to java-centric) functionality and by choosing this feature you can see/analyze the projects created with mainframe resources.

  Note: This feature is selected by default during the installation process of **IBM AD Analyze Client** into **IBM IDz**
- **Application Discovery integration with Common Access Repository Manager (CARMA)** - By choosing this feature you can view the contents of the resources in the Editor view (directly from mainframe, and not from the disk repository used by the IBM AD server) when you work with mainframe resources residing in Endeavor. To set up a CARMA connection, you need to have the CARMA feature installed either on the RDz or IDz platform. This feature requires to have an RSE connection to the mainframe and use TSO credentials when viewing the mainframe-based source assets.
- **Application Discovery integration with IBM CICS Explorer** - By choosing this feature you have a new context menu action **Application Discovery** > **Analyze** available when a **Transaction** is selected in **Transactions** views from **CICS Explorer**. After you select a mainframe project from the list and the analysis to be performed (for example, Transaction Callgraph), the analysis will be displayed. This feature requires **IBM CICS Explorer SDK**.

- **Application Discovery integration with IBM CICS IA** - By choosing this feature you can open an Interdependency Analyzer connection. You can select program names from various CICS IA views (for example, Programs, Command flow) and from the new context menu action **Application Discovery** > **Analyze** , you can start an analysis in your IBM AD Analyze Client IDz instance.
- **Application Discovery integration with IBM Data Source Explorer** - By choosing this feature you have a new context menu action **Application Discovery** > **Analyze** available when a table is selected in **Data Source Explorer**. After you select a mainframe project from the list and the analysis to be performed (for example, SQL Table Usage), the analysis will be displayed.
- **Application Discovery integration with IBM Explorer for z/OS** - By choosing this feature you can view the contents of the resources in the Editor view (directly from mainframe, and not from the disk repository used by the IBM AD server) when you work with mainframe resources in ChangeMan ZMF, and also of type PDS. This feature requires to have an RSE connection to the mainframe and use TSO credentials when viewing the mainframe-based source assets.
- **Application Discovery integration with NPS Survey** - By choosing this feature you can offer feedback in **IBM AD Analyze Client** that is installed in **IBM Developer for z Systems**.

  Note: This feature is selected by default during the installation process of **IBM AD Analyze Client** into **IBM IDz**
- **Application Discovery java analysis** - By choosing this feature you can create and analyze projects made up of Java assets. Examples of such analysis: Resource Dependencies Graph, Java Call Graph. For more information, see “Legacy Java Graphs” on page 354.

  Note: In multi-user installations, Java projects can be shared and registered on an **IBM AD Analyze Server**.
- **Application Discovery java/mainframe cross analysis** - By choosing this feature you can view the cross analysis between mainframe projects and Java projects.

Installing IBM AD Connect for Mainframe

IBM AD Connect for Mainframe is distributed in SMP/E format. To install the latest version, perform the following steps:

1. Follow the instructions found in the Installing IBM Application Discovery Connect for Mainframe chapter from the *Program Directory* document. It is a PDF included as part of the zipped installation package.
2. After installing via SMP/E, copy the load modules contained in the SMP/E *.SIAYAUTH library into the authorized load library from which the listener started task will run.
3. Copy the IAYLSTNR member from *.SIAYSAMP library into the procedure library from which you will run the listener started task. The listener PROC name for the started task could be any name that fits your site’s naming conventions.

For more information regarding the installation process, see the **Installing IBM Application Discovery Connect Mainframe** chapter from the Program Directory for IBM Application Discovery Connect for Mainframe document.

Note: The **Program Directory for IBM Application Discovery Connect for Mainframe** document it is also provided in .PDF format and it is included in the zipped installation package.

After IBM AD Connect for Mainframe has been installed, please perform the following configuration steps, which apply to all versions:

1. Configure the listener PROC as detailed in “Configuring the Listener PROC” on page 152.
2. Configure the needed JCLs as detailed below (as needed).
3. Define a User Id in RACF (with the correct authorizations) for the listener started task.
4. If Db2 access is needed, submit the IAYDB2BD (formerly EZLDB2BD) job. Remember that this is a sample job that has to be modified according to your site standards.
5. Create IBM AD Parm Library if needed (will contain the components sample jobs).

6. If ChangeMan ZMF is needed, add the XMLREQJB job to the Parm library and modify it according to your site standards.
7. If Adabas is needed add the IAYNATDB job (formerly NATDBJOB) to the Parm library and modify it according to your site standards.
8. If Natural is needed, add the IAYNATLB job (formerly NATLBJOB) and IAYNATMM job (formerly NATMMJOB) to the Parm library and modify it according to your site standards.
9. Start the listener started task. Make sure that all required libraries defined in the STEPLIB (or otherwise defined in the linklist, etc.) are APF authorized, as this is required to ensure the integrity of program call chains. Also, ensure that the user id assigned to the started task has the correct authorizations.

Files included in the IBM AD Connect for Mainframe distribution

The following files are included:

- **IBM.HALT610.SMPMCS.BIN** – a compressed SMP/E MCS file, which acts as an inventory for all the software objects included in the SMP/E distribution.
- **IBM.HALT610.F1.BIN** – a compressed SMP/E REL file, containing all the load modules for IBM AD Connect for Mainframe.
- **IBM.HALT610.F2.BIN** - a compressed SMP/E REL file, containing all the Db2 DBRM members. These are required only if IBM AD Connect for Mainframe must connect to Db2 to retrieve application data.
- **IBM.HALT610.F3.BIN** – a compressed SMP/E REL file, containing all the sample JCL and PROCS needed to run IBM AD Connect for Mainframe. For a list of the samples, see “Sample JCL and PROCS for IBM AD Connect for Mainframe” on page 525.
- **IBM.HALT610.F4.BIN** – a compressed SMP/E REL file, containing compiled REXX modules for the optional REXX integration feature.

Chapter 4. Upgrading

Important: Once the upgrade and configuration of IBM Application Discovery and Delivery Intelligence to the latest version is completed, make sure to reactivate the full functionality of **IBM AD** by using the latest Activation Kit available on **Passport Advantage**. For more information, see “ Activating the full functionality of IBM AD from a free version ” on page 461.

© Copyright IBM Corp. 2010, 2022

Chapter 5. Configuring

After IBM AD is installed, you can configure the IBM AD product as follows:

- By using the **IBM AD Configuration Wizard**. For more information, see "Configuring by using IBM AD Configuration Wizard" on page 33.

  IBM AD Configuration Wizard helps to easily configure the Application Discovery servers, databases, and components, by using the **default settings**.
- By using the new interface of **IBM AD Configuration Service**. For more information, see "Advanced Configurations" on page 36.

  After the installation and configuration of IBM AD, you can use **IBM AD Configuration Service** to easily perform **advanced configurations** on the Application Discovery servers, databases, and components, without being necessary to manually update the configuration files. To access **IBM AD Configuration Service**, go to **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**.
- By following the "Manual Configurations" on page 54 steps.

  Important: It is recommended to follow the manual configuration steps only in case that you performed an upgrade to the latest version of **IBM AD** product.

Configuring by using IBM AD Configuration Wizard

IBM AD Configuration Wizard helps to easily configure the Application Discovery servers, databases, and components, without the need to manually update the configuration files.

Important:

- **IBM AD Configuration Wizard** is available only for clean installations, where the configuration of all IBM AD components is currently done without using **IBM AD Authentication Server (DEX)**. To simplify the configuration process, all IBM ADDI components must be installed on the same machine.
- If you use **Db2 for LUW (Linux, UNIX, and Windows)** you must **manually** create the Audit database before you run the IBM AD Configuration Wizard. For more information, see "IBM AD Web Services Prerequisites" on page 17.
- If you use **Db2 for z/OS**, you must **manually** create the Audit database before you run the IBM AD Configuration Wizard or define a relational database server against Db2 for z/OS. For more information, see "IBM AD Web Services Prerequisites" on page 17.

1. If during the installation process of the **IBM ADDI** application, the **IBM Application Discovery Configuration Wizard** check box was selected, the **IBM Application Discovery Configuration Wizard** page will open in the system's default browser after the installation is completed.

   Note:

   - If you need to use **IBM Application Discovery Configuration Wizard** to perform the initial setup at a later time, access https://localhost:9443/ad/admin/setup. It is recommended to use **IBM Application Discovery Configuration Wizard** only once, right after the installation process of **IBM ADDI** product is completed.

     Note: In case that you cannot access https://localhost:9443/ad/admin/setup, as an alternative, you may access http://localhost:9080/ad/admin/setup.
   - **IBM AD Configuration Wizard** is now multi-language ready. Once you access its web interface, you can select the language menu from the upper right corner and choose one of the available languages. The entire configuration process will be displayed in the selected language.
2. On the **Welcome** page, click **Get Started!**.
3. On the **General Settings** page, perform the following actions:

© Copyright IBM Corp. 2010, 2022

- Select the **Encoding**. The default value of the encoding that is used in the source files is **windows-1252**. If necessary, the encoding can be changed. For more information, see https://www.iana.org/assignments/character-sets/character-sets.xhtml.
- Select or enter the **IP address** of the machine where the **IBM ADDI** product is installed.
- In the **Ports** section, all default ports are present. If needed, the ports can be changed.
- In the **Create Environment** section, enter the name of the environment or you can leave it as **Default**.

  Note: When a new environment is created, a default workspace is also created and automatically attached to the new environment. The default workspace is where all **IBM AD Build** projects are available.

4. On the **Paths** page, enter the following information:
   - **Default projects path** - enter the path where all mainframe projects will be stored.
   - **Path for the retrieved members** - enter the path where source code files that are automatically downloaded from Mainframe systems (by using **IBM AD Connect for Mainframe**) will be stored.
   - **z/OS Configurations path** - enter the path where all z/OS configurations will be stored.
   - **Manual Resolutions path** - enter the path where all journal files will be stored.
   - **Search indexes path** - enter the path where all indexes will be stored.
5. On the **Relational Database Server** page, enter the following information:
   - **Server name** - enter an appropriate name (alias) for the relational database server.

     Note: For a Db2 for LUW (Linux, UNIX, and Windows) database server, the server name must follow the general Db2 object naming rules. This alias will be used to catalog Db2 for LUW server node into local node directory in case that you decide automatically create the databases within IBM ADDI (Project database, Cross Service database, Annotation database).
   - **Select database server type** - Select the appropriate relational database type. If you select the **IBM Db2® for z/OS** option, the **Storage Group** field is displayed as optional. In the **Storage Group** field, specify the storage group that is configured for AD on the Db2® for z/OS® server.
   - **Host** - enter the IP or the hostname where the relational database server is installed.
   - **Port** - enter the relational database server port. The default port for SQL Server is 1433.
   - **Instance/location (optional)** - If Microsoft SQL Server was selected as **database server type** and multiple instances are installed on the same machine, enter the required relational database server instance name (if exists).
   - **Username** and **Password**

     a. For Microsoft SQL Server instance enter the username and password for the IBM AD SQL Identity as previously defined in “Microsoft SQL Server Configurations” on page 13.

     b. For IBM Db2 for z/OS instance enter the username and password as previously defined in “IBM Db2 for z/OS Server Configurations” on page 14.

     c. For Db2 for LUW (Linux, UNIX, and Windows) instance enter the username and password.

   Important: By clicking **Test Connection**, you can test the connection to your database. If the test connection succeeds, click **Next**.
6. On the **Database Names** page, enter the following information:

   Note: The names of the databases need to respect the database name restrictions that are in place on the type relational database server (SQL, Db2®) that was associated.
   - Enter the **Annotations Database** name and schema name.
   - Enter the **Audit Database** name.
   - Enter the **Cross Analysis Database** name.
7. On the **Graph Database Server** page, set a new password for GraphDB Server's root user and confirm it.

8. On the **Batch Server** page, the following information is available:
   - **Projects** - the asterisk ***** means that all projects are considered for this service.
   - **Components** - represents a list of component names that are initially selected for this service.

     Note: Additional components can be considered for this service. For more information, see the Additional Components table.
9. On the **Summary** page, click **Commit**.
10. When the configuration is successfully completed, click **Done**

You are redirected to the new **IBM AD Configuration Service** page from where you can perform the "Advanced Configurations" on page 36.

Configuring by using IBM AD Configuration Service

After the installation and configuration of IBM AD, you can use **IBM AD Configuration Service** to easily perform **advanced configurations** on the Application Discovery servers, databases, and components, without being necessary to manually update the configuration files. To access **IBM AD Configuration Service**, go to **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**.

IBM AD Configuration Server ensures that the installation parameters are consistent throughout the different components of IBM AD by storing them in a central location, in a scalable and fail-safe manner .

IBM AD Configuration Server also allows the system administrator to coordinate the access to the resources by creating workspaces and user groups.

When dealing with a suite of applications that work together over a network, a problem that arises often is how to identify applications that belong to separate installations. Simply deploying every parallel installation to separate networks is not always possible, therefore, the applications belonging to the same installation must be able to locate each other and the accessible resources without interfering with applications from other installations. In **IBM AD Configuration Server**, such an installation is called "environment". One **IBM AD Configuration Server** instance can manage one or several environments.

A workspace is a collection of projects to which the administrator can grant access to the users of **IBM AD**. Within an environment, the administrator can create several workspaces. A workspace can contain several related projects, as, for example, when one needs to generate a cross analysis against several projects, one can group these projects in a workspace.

User groups can be defined in **IBM AD Configuration Server**. The administrator can then grant access rights to a specific workspace for a group of users. User groups can be populated by adding individual users or by importing user lists.

Users can be either added individually or imported from user lists.

Important:

- By default, **IBM AD Configuration Server** cannot be opened and used from another machine. To remotely access **IBM AD Configuration Server** go to `<IBM ADDI Installation Folder>/IBM AD Web Services/wlp/usr/servers/ad_server/conf.audit/audit.properties` and change the value of the `com.ibm.ad.audit.security.ui_remote_access` parameter to *true*. Example:

  ```
  com.ibm.ad.audit.security.ui_remote_access=true
  ```

- By default, **IBM AD Configuration Server** is using `http` transfer protocol and thus enabling remote access to **IBM AD Configuration Server** it could expose the information over the network.

The **IBM AD Configuration Server** is now multi-language ready. Once you open its web interface, you can select the language menu from the top right corner and choose one of the available languages. The entire application will be displayed in the selected language.

Advanced Configurations

Filling the IBM AD Build Client Install Configuration Page

In order for **IBM AD Build Client** to work properly, make sure that all the required paths are present in the **IBM AD Build Client Install Configuration Page** before "Adding an Environment" on page 38.

1. To update the **IBM AD Build Client Install Configuration Page**, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Install Configurations** > **IBM Application Discovery Build Client Install Configuration**.
2. The **IBM Application Discovery Build Client Install Configuration** page is displayed.

   If necessary, modify the following information:

   Default project path
   Modify the default path where all **IBM AD Build Client** projects are stored. The path needs to be accessible for any of the following IBM AD components: **IBM AD Build Client** and **IBM AD Batch Server**. This default path can be also changed when you create a project in **IBM AD Build Client**.

   zOS configuration folder
   Modify the default path where the z/OS® Connections are stored. The path must be accessible for any of the following IBM AD components: **IBM AD Build Client** / **IBM AD Build Configuration**.

   Path for the retrieved members
   Modify the default path where all the members that are downloaded from a Mainframe system are stored. The path needs to be accessible for any of the following IBM AD components: **IBM AD Build Client** and **IBM AD Batch Server**.

   Check the following options:

   Enable members synchronization
   If the **Enable members synchronization** option is checked you need to enter the default path of the configuration file that contains the parameters for the synchronization process in the **Path for members synchronization configuration file** field. For more information, see "Synchronizing Mainframe Members" on page 266.

   Enable communication logging
   You can decide whether the communication logs are saved for later use and inspection.

   Keep communication buffers
   You can decide whether the communication buffers are saved for later use and inspection.

Filling IBM ADDI Extension Install Configuration Page

Learn how to specify the configuration properties in the **IBM ADDI Extension Install Configuration Page**.

1. To update the **IBM ADDI Extension Install Configuration Page**, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Install Configurations** > **IBM Application Discovery and Delivery Intelligence Extension Install Configuration**.

   The **IBM Application Discovery and Delivery Intelligence Extension Install Configuration** page is displayed.
2. Specify the following configuration properties in the **IBM ADDI Extension Install Configuration Page**.

Table 1. Properties in the IBM Application Discovery and Delivery Intelligence Install Configuration page

Tab	Name	Description
Web and Application Server	Base URL	The base URL of the IBM ADDI Extension server without any path information. For example, if you want to reach IBM ADDI Extension with a url such as `https://addi.mycompany.com:9753/addi/web` in the future, enter the url `https://addi.mycompany.com:9753`. Make sure that the IBM ADDI Extension server machine is configured with this name and the fully qualified name is configured in your DNS so that it can be reached inside your organization like that. For more information, see Planning URLs. The default port for IBM ADDI Extension is 9443. However, on the production environment where you deploy IBM ADDI Extension with other IBM Application Discovery components, you need to change the port number to avoid conflicts. Because IBM AD Web Services also uses port 9443. Also, the following ports cannot be used since they are occupied by other components: • Port 9444 is used by Elasticsearch Front Server. • Port 7600 is used by Authentication Service (DEX). Before you assign a port to IBM ADDI Extension, make sure that the port doesn't conflict with any other services running on the server. For more information, see “TCP Port Requirements and Firewall Exceptions” on page 20.
Databases	Database server type	You need to create a Data Warehouse (DW) database before configuring IBM ADDI Extension. For an evaluation setup, you can use the Derby by default. The Derby database will be created automatically if it does not exist. If you are setting up IBM ADDI Extension with DB2 for production, select DB2 from the list and provide the details.
	Host	This is the IP or fully qualified network name of your DB2 database server.
	Port	This is the port where DB2 can be reached on the DB2 server. The default port for DB2 is `50000`.
	Database Name	The name of the database. It is recommended to use the name Data Warehouse (DW) for the databases to be created.
	Tablespace Folder	A folder that has been created on the DB2 server machine in which table spaces can be created during setup. The DB2 Admin user must have full write access to this folder. **Note:** If the data warehouse database is used from a previously migrated release of IBM ADDI Extension, the default tablespace from that database will be used regardless of the specified setting of the Tablespace Folder. Because the Tablespace Folder field is mandatory, you can specify anything in case of the default tablespace from the migrated database. If a non-default tablespace was used, then enter that tablespace location in this Tablespace Folder field.
	Username	The username of the DB2 Admin user that was created on the DB2 server machine.

Table 1. Properties in the IBM Application Discovery and Delivery Intelligence Install Configuration page (continued)		
Tab	**Name**	**Description**
	Password	The password for DB2 Admin user. **Note:** This password also needs to be specified when you run the `adi-setup` application.
Authentication Services	Host	This is the IP or fully qualified network name of Authentication Server (DEX).
	Port	The port that is used for the Authentication Server.
	HTTP protocol	The protocol configured for this Authentication Server. Select HTTP or HTTPS.
User Groups	Admin Group List	Enter a comma-separated list of user groups that will have *Administrator privileges* on the configured IBM ADDI Extension server.
	User Group List	Enter a comma-separated list of user groups that will have *User privileges* on the configured IBM ADDI Extension server. This list of user groups will be made available for each IBM ADDI Extension workbook such that the admin can select a sublist to provide them access to this workbook.

Adding an Environment

Learn how to add an environment to a configuration server in **IBM AD Configuration Server**.

1. To add a new environment in **IBM AD Configuration Service**, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments**. The **Environments** page is displayed.
2. Click **Add** and enter the following information:
   - **Name** - enter a name for the new environment.
   - **Description** - enter an appropriate description for the new environment.
3. Click **Save**. The new environment is displayed in the Environments list.

Setting the parameters for the Advanced Configuration

Modify the General and Port Settings

1. To update the **General and Port Settings**, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments**.
2. The **Environments** page is displayed. Select the environment for which you want to modify the General and Port settings.
3. The parameters that can be updated for the selected environment are displayed under the **General** tab.

   If necessary, modify the following information:

   Encoding
   The default value of the encoding that is used in the source files is **windows-1252**. The encoding can be changed. For more information, see https://www.iana.org/assignments/character-sets/character-sets.xhtml.

IP
You can select or enter the **IP address or the hostname** of the machine where the **IBM ADDI** product is installed.

List of default ports used by the following components:

Note: All default ports are set during the installation process through **IBM AD Configuration Wizard**. If needed, you can change the default ports to different ports. For example, in the event of a port conflict, where another application is using these port ranges, you can change the default ports to avoid the conflict.

Audit and Catalog
The default value is 9080.

Zookeeper
The default value is 2181.

Analyze Server
The default value is 1099.

Graph database server
The default value is 2424.

Cross Applications Service
The default value is 7850.

File Service
The default value is 7700.

Mainframe Projects Service
The default value is 7650.

Manual Resolutions Service
The default value is 7900.

Search Service
The default value is 7800.

Configuring the Servers and Security Settings

Managing a Relational Database Server

By adding a relational database server, you correctly configure the connection between IBM AD Build and IBM AD Analyze. This task is step 5 of configuring an IBM AD installation through IBM AD Configuration Wizard. You can add, update, or delete a relational database server as follows.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Relational database server**. The **Relational Database Server** settings page is displayed.
2. To add a relational database server, click the **Add** button. The **Add Relational Database Server** page is displayed.
3. Enter the following information:

   Server Name (alias)
   Enter an appropriate name (alias) for the relational database server.

   Note: For a Db2 for LUW (Linux, UNIX, and Windows) database server, the server name must follow the general Db2 object naming rules. This alias will be used to catalog Db2 for LUW server node into local node directory in case that you decide automatically create the databases within IBM ADDI (Project database, Cross Service database, Annotation database).

   Select Database Server Type
   Select the appropriate relational database type.

Host
Enter the host name or the IP of the computer where the relational database server is installed. IBM AD Build Configuration uses the server to publish projects.

Port
Enter the relational database server port. The default ports are:

For Microsoft SQL Server
The default port for Microsoft SQL Server is 1433.

For Db2 for z/OS
The default port for Db2 for z/OS is 50000.

For Db2 for LUW (Linux, UNIX, and Windows)
The default port for Db2 for LUW when TLS is **disabled** is 50000.

Use TLS



Attention:

- The **Use TLS** check box is used to enable Transport Layer Security (TLS) communication with Db2 for LUW and it is available **only** when **IBM Db2** is selected as the database server type. To be able to select the **Use TLS** check box make sure that **Hypertext transfer protocol secure (HTTPS)** is set as a default connection protocol in **IBM AD Configuration Server**. For more information, see "Configuring Security Settings" on page 42.
- For the complete steps on how to configure DB2 for LUW with TLS, see Securing IBM Db2 for LUW database connections.

Database Name (optional)
If **IBM Db2** is selected as the database server type, the **Database Name (optional)** field is displayed. You can enter the name of any database on the selected server to validate the connection. This field is optional.

Storage group (optional)
If **IBM Db2 for z/OS** is selected as the database server type, the **Storage Group (optional)** field is displayed. Specify the storage group that is configured for IBM AD on the Db2 for z/OS server. This field is optional.

Instance (optional)
If **Microsoft SQL Server** is selected as the database server type and multiple instances are installed on the same machine, enter the required relational database server instance name (if exists).

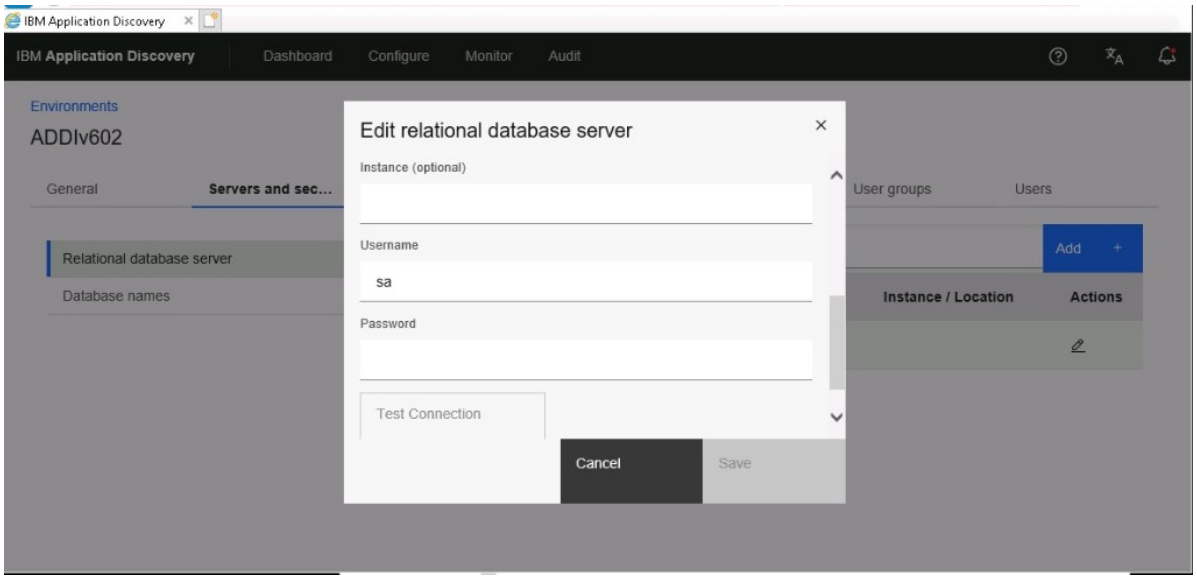
Username and Password

a. For Microsoft SQL Server instance, enter the username and password for the IBM AD SQL Identity as previously defined in "Microsoft SQL Server Configurations" on page 13.
b. For IBM Db2 for z/OS instance, enter the username and password as previously defined in "IBM Db2 for z/OS Server Configurations" on page 14.
c. For Db2 for LUW (Linux, UNIX, and Windows) instance, enter the username and password.

Note: If you need to update the name of the AD SQL Identity user for your Microsoft SQL Server environment, complete the following steps:

a. Create a new SQL Identity user in SQL Server Management Studio, and grant the same permissions to the new user as the existing SQL Identity user. For the detailed steps, follow the description in "Microsoft SQL Server Configurations" on page 13.
b. After you define the new AD SQL Identity user in SQL Server Management studio, return to the IBM Application Discovery Configuration Service Admin graphical user interface in the web browser. Then, click **Relational database server** tab, and click the pencil icon under the **Actions** tab to edit an existing relational DB server definition.

c. After the **Edit relational database server** page is displayed, scroll to the **Username** and **Password** fields, and update the name of the SQL Identity user and password if appropriate in the Relational database server definition in AD's Configuration Server as described at this step.



4. By clicking **Test Connection**, you can test the connection to your database. If the test connection succeeds, click **Save**.
5. To edit a relational database server, go to the **Actions** column and select the **Edit** button.
6. To delete a relational database server, go to the **Actions** column and select the **Delete** button.

Configuring Database Names

This task is step 6 of configuring an IBM AD installation through IBM AD Configuration Wizard. You can edit the database names as follows.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Database names**. The **Database Names** settings page is displayed.
2. Associate a database server from the ones that are already defined under **Relational database server** and add the following information:

   Note: The name of the databases need to respect the database name restrictions that are in place on the type of relational database server (Microsoft SQL Server, Db2 for LUW (Linux, UNIX, and Windows) or Db2 for z/OS) that was associated.

   Annotations Database
   Enter the Annotations Database name and the schema name. The Annotations Database is used to store the comments that are related to application's lifecycle added by a user on a resource.

   Audit Database
   Enter the Audit Database name.

   Cross Analysis Database
   Enter the Cross Analysis Database name.

   Note: You can add the Cross Analysis Database name only when the associated relational database server type is **IBM Db2** or **Microsoft SQL Server**.
3. Click **Save**.

Configuring Graph Database Server

The graph database configuration is part of the IBM AD installation configuration and consists of configuring the connection details for **IBM AD GraphDB** component. If the graph database connection

details are not configured, some of the analyses in **IBM AD Analyze** are not available and some **IBM AD Batch Server** tasks cannot be performed. Configure the connection details as follows.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Graph Database Server**. The **Graph Database Server** settings page is displayed.
2. During the installation process through **IBM AD Configuration Wizard**, the password for GraphDB Server's root user was set. If necessary, set a new password and confirm it.
3. Click **Save**.

Configuring Security Settings

By default, the endpoints available on the ports of **IBM AD File Service**, **IBM AD Search Service**, **IBM AD Mainframe Projects Service**, **IBM AD Cross Applications Service**, **IBM AD Manual Resolutions Service**, **IBM AD WebSphere® Liberty Profile Service**, and **Authentication Server (DEX)** are publicly available inside the domain, where no access control is enforced.

You can enable **Hypertext transfer protocol secure (HTTPS)** as a default connection protocol as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Security**. The **Security** settings page is displayed.
2. Select **HTTP Secure (https)** as a default connection protocol.

   Important: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`), a non-encrypted private key for the certificate (`.key`), and the keystore file that needs to have one of the following extensions: `.jks`, `.keystore`, `.pfx`, `.p12`, or `.ks`.
3. Drag and drop the three required files or click to browse.
4. After all files are successfully uploaded enter the **Keystore Password**.
5. Click **Save**.
6. After **Hypertext transfer protocol secure (HTTPS)** was set as a default connection protocol you can enter an encrypted **Keystore Password**. For more information, see "Obtaining an encrypted keystore password" on page 209.

You can set up a secure communication for:

- **IBM AD Batch Server and IBM AD Graph Database Server**

  For more information see "STEP 8. Configuring IBM AD Batch Server" on page 75 (without authentication) or "STEP 8. Configuring IBM AD Batch Server" on page 124 (with authentication).
- **IBM AD Configuration Server**

  For more information, see "Setting up IBM AD ZooKeeper to run in Mixed Mode" on page 203.
- **IBM AD Analyze Client**

  For more information, see "Enabling encryption channel between IBM AD Analyze Client and IBM AD ZooKeeper" on page 208.

Configuring Authentication Server (DEX)

The authentication process is conducted through **DEX** or through any other **Authentication Server** that supports the **OAuth2** protocol. **Authentication Server (DEX)**, which is delivered through the **IBM ADDI** installer, is an identity service that uses **OpenID Connect** and it is used in the following configurations.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Authentication Server (DEX)**. The **Authentication Server (DEX)** settings page is displayed.

2. If you want to use **Authentication Server (DEX)**, make sure that the **Enable Authentication** check box is selected.
3. In the **General** section, enter the following information:

   Authentication server hostname
   Expects the authentication server hostname that is used by Authentication Server (DEX). Example:

   ```
   example.com
   ```

   Callback URL
   Expects a URL composed by localhost IP and a generic port that is used for callback to **Authentication Server (DEX)**. You can leave the default value.

   Timeout (in seconds)
   Expects the time for the client to respond. The default value is **120**.
4. In the **Active directory** section, configure the authentication server (DEX) to use an LDAP server to act as a user registry and pull user and user group information from it.

   LDAP registry location hostname
   Expects the hostname of the LDAP registry. Example:

   ```
   example.com
   ```

   LDAP registry location port
   Expects the port of the LDAP registry. Default value: 389. If the **SSL** implementation is used, the default port of the LDAP registry is 636.

   Username (optional)
   Expects the account name that is used to access the LDAP registry.

   Note:

   - Leave empty if the LDAP allows anonymous access to retrieve user and user group information.
   - To add the account that has rights for LDAP bind action run `adsiedit.msc` on the Active Directory machine and load the current domain. Right click on `CN=Users` and `CN=Administrator`, select `Properties` and search for `distinguishedName` attribute. For more information, see ADSI Edit (adsiedit.msc).

   Password (optional)
   Expects the account's password that is used to access the LDAP registry.

   Note: Leave empty if the LDAP allows anonymous access to retrieve user and user group information.
5. In the **User search** section, specify the following properties to configure the authentication server (DEX) to map to users in the LDAP server.

   Base user DN

   Expects the base distinguished name of the users in the LDAP registry. Example :

   ```
   CN=UserContainer,DC=Domain,DC=com
   ```

   Filter
   You can leave the default value. Example:

   ```
   (objectClass=person)
   ```

   Username
   You can leave the default value. Example:

   ```
   userPrincipalName
   ```

   Email attribute
   You can leave the default value. Example:

```
userPrincipalName
```

6. In the **User group search** section, specify the following properties to configure the authentication server (DEX) to map to user groups in the LDAP server.

   Base Group DN

   Expects the base distinguished name of the groups in the LDAP registry. Example:

   ```
   CN=GroupContainer,DC=Domain,DC=com
   ```

   Filter

   You can leave the default value. Example:

   ```
   (objectClass=group)
   ```

   Group member attribute

   Expects an attribute to represent the members of a group in the LDAP registry. Example: `member`.

7. Click **Save**.

Configuring Reports

The Reports configuration consists of determining what information is displayed on the first page of the reports that are generated in **IBM AD Analyze Client**. Also, you can set the company logo that is to be used in all the generated reports.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Reports**. The **Reports** settings page is displayed.
2. In the **Owner** field, enter the name of the user who generates the reports.
3. In the **Organization** field, enter the name of your organization.
4. In the **Organization role** field, enter the role of the user within the organization.
5. If the **Activate view** option is selected, in **IBM AD Analyze Client**, AD Reports view is displayed automatically on top of the current view when the report generation process is finished. If this option is not selected, the user has to check AD Reports view to see if the report generation process is finished.
6. Drag and drop an image file or click to browse for the file that contains the logo image. `*.PNG` and `*.GIF` formats are supported. A preview of the selected file is displayed.
7. Click **Save**.

Configuring Rule Based

The Rule based configuration consists loading a file that contains the rules to be used when the **Rule based reports** are generated in **IBM AD Analyze Client**. The Rules file can be loaded either from **IBM AD Configuration Server** or from **IBM AD Analyze Client**. Also, the custom rules encoding can be specified. An example of a Rules file is available in *IBM AD Analyze Client and Server User Guide*, in *Appendix 2: Rules Files Example for Rule Based Analysis*.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Rule Based**. The **Rule Based** settings page is displayed.
2. Select one of the **Custom rule encoding** options available.
3. Drag and drop a `*.zip` file or click to browse for the custom rules file archived in `*.zip` format.
4. Click **Save**.

Configuring the Services Settings

Configuring IBM AD Mainframe Projects Service

Through **IBM AD Mainframe Projects Service** a user is authorized to access the AD projects.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Mainframe Projects Service**. The **IBM AD Mainframe Projects Service** settings page is displayed.
2. Set the **Poll interval** parameter to determine how frequently the projects are tested (seconds).
3. Click **Save**.

   Note: Make sure to restart **Mainframe Projects Service** after saving the changes.

Configuring IBM AD Cross Applications Service

By adding a restriction in the **Cross Applications Service** page, you can decide what project resolutions to display in the analysis, when more projects contain the same program definition.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Cross Applications Service**. The **Cross Applications Service** settings page is displayed.
2. To add a new restriction, click the **Add** button. As a result, the **Add Restriction** page is displayed.
3. In the **Pattern** field, enter the name of the program that has the source present in other projects.
4. In the **From Project** field, choose the project that contains the program without the source
5. In the **To Project** field, choose the project resolution that you decide to show in the analysis.
6. Click **Save**.

   Note: Make sure to restart **Cross Applications Service** after saving the changes.

Configuring IBM AD File Service

The access rights of users or users' groups are mapped to a certain folder that contains the source files. A user can start the analysis on the source files if it has read access rights.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **File Service**. The **File Service** settings page is displayed.
2. In the **Mappings** section, click the **Add** button to add a new mapping and enter the following information:
   - **Remote** - add the remote path that can be used to query this service, for example, a UNC path or a local path where the resources/projects reside.
   - **Local** - add the local path that mirrors the remote path. If missing, it is identical to **Remote**.

     Note:

     – In case you have multiple physical folders that reside resources or projects, an entry for each folder needs to be added in the **mapping** section.

     – In order to see **Flow Chart** analysis in **IBM AD Analyze Client** for a specific project, you need to add the path of the project folder or the path of the folder that contains all projects.

       Examples:

       - `\\9.20.128.222\Projects` - path to all projects.

       - `\\9.20.128.222\ADProject` - path to a specific project.
   - Define which **User Groups** can access the related mapping folders.

     Note: If no **User Groups** are defined, all users will have access to the related mapping folders.
   - Click **Save**.
3. In the **Group Settings** section, enter the following information:

- **Session Timeout** - add the session timeout to determine how long the service needs to remain connected to **IBM AD Configuration Server** before it must perform a new connection. The value format must be a whole number followed by 's' for seconds, 'm' for minutes or 'h' for hours.
- **Groups Polling** - add the refresh time to check periodically the groups that are present in **IBM AD Configuration Server**. The value format must be a whole number followed by 's' for seconds, 'm' for minutes or 'h' for hours.

4. Click **Save**.

   Note: Make sure to restart **File Service** after saving the changes.

Configuring IBM AD Search Service

IBM AD Search Service is responsible with the access to the indexed data. The folder path in which the indexes are generated needs to be accessible for both **IBM AD Batch Server** and **IBM AD Search Service**. This task is step 4 of configuring an IBM AD installation through **IBM AD Configuration Wizard**

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Search Service**. The **Search Service** settings page is displayed.
2. Add the path where the **indexes** are created. This path needs to have as an endpoint the same folder where the **indexes** are stored. Example:

   ```
   \\server01\Indexes
   ```

3. Optionally, the main path where the indexes are created for each project, can be overwritten. Click **Add**, select the target project, and add the path where the indexes are created for the target project.
4. Click **Save**.

   Note: Make sure to restart **Search Service** after saving the changes.

Configuring IBM AD Manual Resolutions Service

The **IBM AD Manual Resolutions Service** manages the manually added resolutions and it is mandatory to be configured if you want to use callgraph-based analyzes (graphs or reports). This task is step 4 of configuring an IBM AD installation through **IBM AD Configuration Wizard**.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Manual Resolutions Service**. The **Manual Resolutions Service** settings page is displayed.
2. If necessary, you can modify the path where the **journal files** are created. The path where these files are generated is separated from the project's path and needs to be accessible only for **IBM AD Manual Resolutions Service**. Once a project is imported, a folder with the same name is generated in the related path and it hosts all the files that are needed to manage dynamic call resolutions. Example:

   ```
   C:\AD\Resolutions
   ```

3. Optionally, the main path where the manual resolutions are created for each project, can be overwritten. Click **Add**, select the target project, and add the path where the manual resolutions are created for the target project.
4. Click **Save**.

   Note: Make sure to restart **Manual Resolutions Service** after saving the changes.

Configuring IBM AD Batch Server Service

Before running **IBM AD Batch Server**, some preliminary configurations must be performed. You need to specify on which projects you want **IBM AD Batch Server** to run the reports, which reports to generate and where to store the generated reports.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your**

environment" > **Services** > **Batch Server Service**. The **Batch Server Service** settings page is displayed.

2. Add the folder path in which the indexes are generated.
3. Select the projects on which **IBM AD Batch Server** will operate.
4. Select at least one component that will run on the selected projects.

   Detailed information on the additional components that can be considered for this service.

Table 2. Additional Components

Component	Description
RuleBased	The **RuleBased** component generates reports for the resources that are specified in the configuration files according to the rules and parameters that are defined in the corresponding configuration files. **Note:** If the **RuleBased** component is used, make sure that the `ruleBased.properties` file is configured. For more information, go to "ruleBased.properties File" on page 87.
Reports	The **Reports** component is used to generate the complexity reports. For more information, go to "Complexity Reports" on page 359.
CobolPreprocessor JCLPreprocessor PL1Preprocessor	The **CobolPreprocessor**, **JCLPreprocessor**, and **PL1Preprocessor** components generate the expanded sources for Cobol, JCL, and PL/I. For more information, go to "View Expanded Source" on page 318.
wsmetrics	The **wsmetrics** component is needed only if IBM ADDI Extension is used on the system. **Note:** Additionally, in order for the **wsmetrics** component to be executed, make sure that the **GraphDatabaseImport** component is included in the components list of this service.
ADDIExtension	The **ADDIExtension** component is needed only if IBM ADDI Extension is used on the system. The ADI Index component will index the resources of a project so that the Business Rules Discovery (BRD) feature can display code snippets.

5. Click **Save**.

   Note: Make sure to restart **Batch Server Service** after saving the changes.

Configuring IBM AD Catalog Service

Learn how to configure **IBM AD Catalog Service** in **IBM® AD Configuration Server**. If you want to analyze APIs and services defined in **z/OS® Connect** you should specify here the settings to access Catalog's webservice to be used in **IBM AD Analyze Client**.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Catalog Service**. The **Catalog Service** settings page is displayed.
2. Set **HTTP Protocol** to determine whether a non-secured or secured communication is used.
3. Set **Authentication**. If the **Basic authentication** option is checked, you need to enter the Username and Password and to confirm the entered password.
4. Click **Save**.

   Note: Make sure to restart **Catalog Service** after saving the changes.

Adding a Workspace

Learn how to add a workspace to your configuration server. All new projects will be added to the default workspace.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Workspace**. The **Workspace** page is displayed.
2. Click **Add** and enter the following information:
   - **Name** - enter a name for the new workspace.
   - **Description** - enter an appropriate description for the new workspace.
   - **Accessibility** - select whether the workspace will be public, closed, or available only for specified user groups. If you select the **Access Control List (ACL)** option you can then select the user groups that have access to the current workspace. Click **Save**. The new workspace is added to the list of available workspaces.
3. Click **Save**.

Assigning a Default Workspace to an Environment

When a new environment is created a Default workspace is automatically created and attached to the new environment. But you can assign a different default workspace for a specific environment.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Workspace**. The **Workspace** page is displayed and contains a list of all the workspaces that are defined in the current environment.
2. Click the **Edit** button of target workspace. The **Edit Workspace** page is displayed.
3. You can choose to have it as a default workspace by setting the **Make default** to **Yes**.
4. Click **Save**.

Adding a Project to a Workspace

Make sure that the **IBM AD Build** projects were published into **IBM AD Configuration Server**. For more details, see *IBM AD Build User Guide*.

Projects can be grouped into workspaces according to security or production requirements. Projects can be either automatically published to a workspace or manually added.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Workspace**. The **Workspace** page is displayed.
2. Click the **Edit** button of target workspace, to add a project. The **Edit Workspace** page is displayed.
3. A list of projects that are already assigned to the current workspace is displayed. You can add a project that is not already assigned to the selected workspace, by clicking **Add**.
4. Select all projects that you want to be assigned to the selected workspace, then click **Save**.

Enabling Business Rules Discovery

The BRD feature allows IBM AD to save data that is used to define Business Rules within IBM ADI.

Note: The user can choose to define the Business Rules for projects that are created in IBM AD Build Client by following this procedure.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Projects**. The **Projects** page is displayed.
2. You can enable the **Enable Business Rules Discovery** for a specific or multiple projects as follows:

a. Click the **Edit** button of target project, select the **Enable Business Rules Discovery** check box and click **Save**.

   Important: Business Rules Discovery was enabled and will come into effect at the next project build. For more information, see “Building Projects” on page 264.

b. Select one or multiple projects and click **Enable Business Rules Discovery** button. As a result, the following message is displayed.

```
Business Rules Discovery was enabled, but it will come into effect at the next project
build.
```

   Note: The same actions can be performed to **Disable Business Rules Discovery**.

Enabling Cross Applications

The Cross Applications feature can be enabled for the projects that are created in **IBM AD Build Client** by following this procedure. The **Cross Applications Callgraphs** can be generated to show calls between different mainframe projects that have their databases on the same DB instance. For more information, see “Cross Applications Callgraph” on page 334.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Projects**. The **Projects** page is displayed.
2. You can enable the **Enable Cross Applications** for a specific or multiple projects as follows:

   Important: The Cross Applications feature **is not supported** if the Relational Database Server Type is **IBM Db2 on z/OS**.

   a. Click the **Edit** button of target project, select the **Enable Cross Applications** check box and click **Save**.

      Important: Cross Applications was enabled and will come into effect at the next project build. For more information, see “Building Projects” on page 264.

   b. Select one or multiple projects and click **Enable Cross Applications** button. As a result, the following message is displayed.

```
Cross Applications was enabled, but it will come into effect at the next project build.
```

      Note: The same actions can be performed to **Disable Cross Applications**.

Adding a User

In order to be granted access to a specific workspace from an environment, a user has to be added in **IBM AD Configuration Server**.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Users**. The **Users** page is displayed.
2. Click **Add user** and enter the following information:
   - **Name** - enter the name of the user.
   - **Domain (optional)** - if required, enter the domain name.
3. Click **Save** to add the user to the system.

Importing a List of Users in Users Page

In order to be granted access to a specific workspace from an environment, a user has to be added in **IBM AD Configuration Server**. You can import multiple users into **IBM AD Configuration Server** as part of the resources management process.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Users**. The **Users** page is displayed.
2. Click **Import**. The **Import users** page is displayed
3. In the **File format** area, the option **Tab Separated Values** is selected by default. **Comma Separated Values** format is also supported. Make the selection according to your needs.
4. Drag and drop a file or click to browse for the file that contains the users list. `*.CSV` and `*.TXT` formats are supported. A preview of the selected file is displayed.

   Note: If you click to browse, a **File Upload** dialog window is displayed. Select the file that contains the users list. Only UTF-8 encoded files are accepted.
5. Based on the header of the CSV/TSV file, a list of available columns is displayed. By selecting the **Automatically detect the domain** option, you can choose the user logon names that have the required `username@domain` format.
6. A list of users that were found in the selected file is displayed showing the selected info. By default, all users are selected for import into the Users page. Review the list and remove the users that must not be imported.
7. Click **Save**. The imported users are displayed under the **Users** page to the left of the screen.

Adding a User Group

By adding a user group, you can determine which users can access the specific workspaces and projects.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **User Groups**. The **Users Groups** page is displayed.
2. Click **Add** and enter the name of the user group.
3. Click **Save** to add the user group to the system.

Adding a User to a Group

Users can be added to user groups to control their access to specific workspaces and projects.

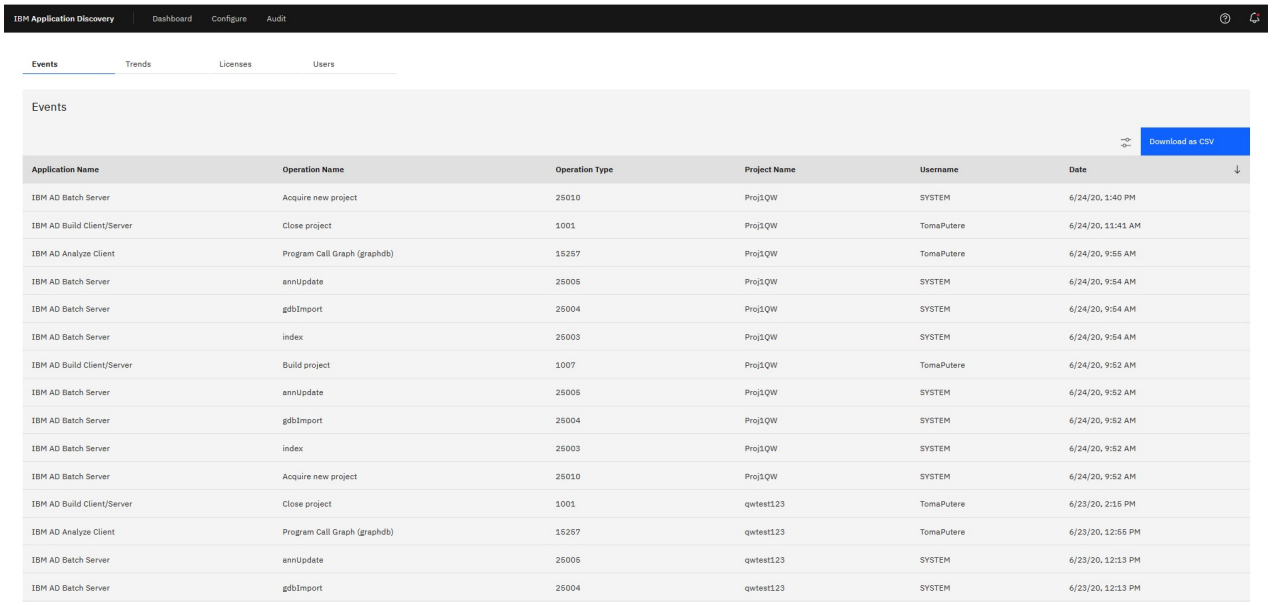
1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **User Groups**. The **Users Groups** page is displayed.
2. Click the **Edit** button of the target user group to add a user. The **Edit User Group** page is displayed.
3. A list of users that can be assigned to the target user group is displayed. You can add a user that is not already assigned to the selected user group, by clicking **Add**.
4. Select all users that you want to be assigned to the selected user group, then click **Save**.

Audit Service Web GUI Overview

With Audit service web GUI, you can monitor the operations and the license usage.

Important: The license monitor capability is not for license compliance purpose.

To access the Audit service web GUI, select **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin** and go to the **Audit** tab. The Audit page is displayed as in the following image.



Notes:

- Use the same port as the one configured in the **AD Web Service**.
- The **Audit** page is available only after **Ad Server** is started and the database connection is set up.

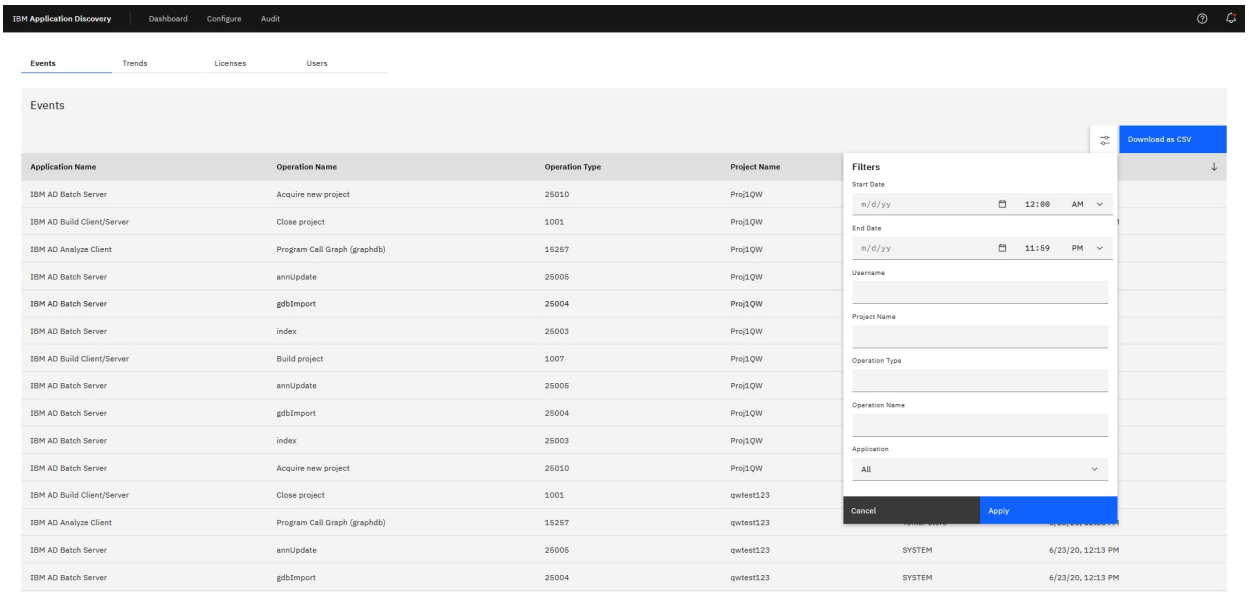
  To set up the database connection, complete the following steps:

  1. Access the Audit service web GUI by selecting **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**.
  2. Select **Configure** > **Environments** > **"Your environment"** > **Servers**.
  3. Specify the settings in the **Relational Database Server** and **Database Names** pages correctly in the active environment.

The Audit page is displayed with the following tabs: **Events**, **Trends**, **Licenses**, and **Users**.

- On the **Events** tab, you can get visibility over all the activities that occurred within IBM ADDI.

  You can use filters to search for specific entries by start and end date, username, project name, operation type and name, or by application. Alternatively, you can click **Download as CSV** to export the results as a `.csv` file.



- The **Trends** tab shows the aggregated license usage in daily, weekly, or monthly intervals. You can compare the real usage with the upper limit that you have. But you can check only the result of latest 90 days.
- On the **Licenses** tab, you can give the number of licenses that you have obtained for the product and set the threshold percentage for the **floating license** usage. Those settings will be used to calculate whether an over usage warning or error notification should be generated or not. In addition, you can define a main list of users who use the **authorized licenses**. By default, everyone uses a floating license.

  The **Authorized User** is a unit of measure by which the Program can be licensed. An Authorized User is a unique person who is given access to the Program. The Program may be installed on any number of computers or servers and each Authorized User may have simultaneous access to any number of instances of the Program at once. The licenses must be obtained separately, dedicated entitlements for each Authorized User given access to the Program in any manner directly or indirectly (for example: via a multiplexing program, device, or application server) through any means. An entitlement for an Authorized User is unique to that Authorized User and may not be shared, nor may it be reassigned other than for the permanent transfer of the Authorized User entitlement to another person.

  The **Floating User** is a unit of measure by which the Program can be licensed. A Floating User is a person who is accessing the Program at any particular point in time. An Install is an installed copy of the Program on a physical or virtual disk made available to be executed on a computer. The Program may be installed on any number of computers or servers, but if the Floating User simultaneously accesses more than one Install of the Program, the Floating User requires a separate entitlement for each such Install. Licensee must obtain separate entitlements for each Floating User simultaneously accessing the Program on each Install in any manner directly or indirectly (for example: via a multiplexing program, device, or application server) through any means.

  A notification will be generated whenever an over usage occurs. It might be a warning or an error. This kind of notifications will be generated only once a day. When a new notification is generated, you can see a red dot on the **Notifications** (bell) icon on the upper right corner of web page as the new notification indicator. Click the **Notifications** icon, a dialog box that shows at most 10 of the latest notifications opens and the red dot disappears.
- The **Users** tab provides a list of users and used licenses. You can assign an authorized user license to a specific user.

Monitoring the Statuses of Components and Projects

On the Monitor page, you can monitor not only the status of IBM AD components but also the running status of projects.

To access the Monitor page, select **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and click the **Monitor** tab on the header. The **Monitor** page is displayed with the following tabs.

- **Components** tab

  By default, you are on the **Components** tab that lists the AD components and shows the relationship between the components through a graphical representation. The following three kinds of components are listed:

  – **Server**: Configuration Server, Batch Server, Graph Database Server, Analyze Server, ADI Server, and Authentication Service

  – **Client**: Build Client, Analyze Client, and Web Client

  – **Service**: Mainframe Projects Service, Web Service, Catalog Service, Manual Resolutions Service, Search Service, Cross Application Service, File Service, Validation Service, Business Rule Discovery Service, and Audit Service

  You can click **Export All Logs** to export the logs of the listed components at once as a `.zip` file.

  For those components that are managed by the Service Manager, you can monitor their statuses directly through the status indicators, that is, Online, Offline, or Unavailable. In addition, you are provided with

options to further work with the components based on your needs. Click the overflow menu (vertical ellipsis) icon on each component card, you can find the following action options from the drop-down menu: **Stop Service**, **Restart Service**, **View Logs**, **Copy Log Folder Path**, and **Edit Configuration**.

- **Projects** tab

  You can create a project and build it from the Build Client. Then, go to this **Projects** tab to see the summary about what is happening in the backend. When the project status is shown as **Ready for Analysis**, you can start working on the project with Analysis Client.

Table 3. Project status data

Data	Description
Name	The **Name** shows the project name.
Status	The **Status** shows the actual status of the project. The available values of the status can be **New**, **In progress**, **Ready for analysis**, **Ready for analysis, partial errors**, or **Fail**. **New** The status indicates that it is a newly created project. **In progress** The status indicates that something is running in the backend, and you must wait until it completes. **Ready for Analysis** The status indicates that the project is ready for analysis. **Ready for analysis, partial errors** The status indicates that the project is ready for analysis, but one or more Batch Server jobs is finished with errors. **Fail** The status indicates that something went wrong, and you need to diagnose and fix the problem.
Build progress	The **Build progress** shows the build status of the project for the build component that includes the following parts. **build type** The **build type** shows what type of the build operation that is being performed, **Build**, **Make**, or **Synchronize members**. **build progress** The **build progress** shows the actual progress of the current build operation. If the part of the build is complete, the actual progress is shown as 100%, and the partial analysis will be available in the Analyze client. Then, after a new build is started later, the progress is reset to 0%. **build duration** The **build duration** shows how long the build operation has been running. It will show that the total build duration after the build progress reaches 100%. Then, after a new build is started later, the progress is reset to 0%.

Table 3. Project status data (continued)	
Data	**Description**
Last build duration	The **Last build duration** shows the duration of the build that was performed. – If it is a new project that is build for the first time, the column is empty. – If it is the second build or all consecutive builds, the duration will be taken from the **Build progress** column before that column is reset to 0.
Batch progress	The **Batch progress** shows the progress of the batch server tasks, that is, the tasks that have completed out of the total number of tasks to be executed, and how long has this been running. The **Batch progress** is the secondary stage of the overall project build that accumulates more project data for extra reports. The stage is ignored for **Synchronize members** and **Make** build type on a project with no files changed since the last build. The performed tasks can be configured from Configuring IBM AD Batch Server Service. After the stage is complete, the **Status** column will be changed to be **Ready for analysis**, which means all tasks completed successfully. Or, the **Status** column will be changed to be **Ready for analysis, partial failures** , which means at least one task is failed to complete. The details of what tasks were performed can be seen on the **Audit** tab.
Last batch duration	The **Last batch duration** shows the duration of the last batch build that was performed. – If it is a new project that is build for the first time, the column is empty. – If it is the second build or all consecutive builds, the duration will be taken from the **Batch progress** column before that column is reset to 0.

Manual Configurations

Important:

Starting with **IBM AD V5.1.0.9 iFix 1**, the following enhancements are available:

- The old UI of **IBM AD Configuration Service** has been removed. The *http://localhost:8080* link cannot be accessed.
- The **Hypertext transfer protocol secure (HTTPS)** for **IBM AD File Service**, **IBM AD Search Service**, **IBM AD Mainframe Projects Service**, **IBM AD Cross Applications Service**, and **IBM AD Manual Resolutions Service** can now be enabled from the new UI of **IBM AD Configuration Service**.

Note:

For any version prior to **IBM AD V5.1.0.9 iFix 1**, it is still recommended to follow the Manual Configuration Steps to enable **Hypertext transfer protocol secure (HTTPS)** as a default connection protocol for **IBM AD File Service**, **IBM AD Search Service**, **IBM AD Mainframe Projects Service**, **IBM AD Cross Applications Service**, and **IBM AD Manual Resolutions Service**.

Without Authentication

STEP 1. Configuring IBM AD Configuration Server

The IBM AD Configuration Server component can run with the default settings. If the default settings are not compatible with your environment, you can configure the component and overwrite the default settings.

1. Configure the settings in the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/server.properties` file.
   a) Configure the port that AD Configuration Server listens on by setting the value of the **`server.port`** parameter.

      The default value is 2181.

   b) Configure the number of the snapshots and the corresponding logs that are retained by AD Configuration Server. To configure the setting, set the value of the **`zookeeper.autopurge.snapRetainCount`** parameter.

      The default value is 4, and the minimum value is 3.

   c) Configure the time interval in hours for the purge task by setting the value of the **`zookeeper.autopurge.purgeInterval`** parameter.

      The purge task deletes old snapshots and the corresponding log files according to the time interval. The default value of the parameter is 24. To enable the purge task, you must set a value that is greater than `0`.
2. Configure the web server settings in the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/webservice.log4j.properties` file.
   a) Configure the root logger level and the appenders by setting the **`log4j.rootLogger`** parameter with one of the following values:

      - `OFF`
      - `FATAL`
      - `ERROR`
      - `WARN`
      - `INFO`
      - `DEBUG`
      - `TRACE`
      - `ALL`

      The default log level is `INFO`. The default appenders are `file`, which indicates a rolling file appender, and `stdout`, which indicates a console appender.

      Example

      ```
      log4j.rootLogger=DEBUG
      ```

   b) Configure the file roller appender log level by setting the **`log4j.appender.file.threshold`** parameter with one of the following values:

      - `OFF`
      - `FATAL`
      - `ERROR`
      - `WARN`
      - `INFO`
      - `DEBUG`
      - `TRACE`
      - `ALL`

Note: If you do not set the value of the **log4j.appender.file.threshold** parameter, the file roller appender log level is the same as the root logger level. To set the **log4j.appender.file.threshold** parameter, the value must be lower than or equal to the root logger level.

Example

```
log4j.appender.file.threshold=ERROR
```

c) Configure the file roller appender location by setting the value of the **log4j.appender.file.File** parameter.

Note: The value of the **log4j.appender.file.File** parameter must be a valid absolute or relative path.

Example

```
log4j.appender.file.File=/home/user/logs/webservice.log
```

d) Configure the file roller appender minimum number of backup files to keep by setting the value of the **log4j.appender.file.MaxBackupIndex** parameter.

Example

```
 log4j.appender.file.MaxBackupIndex=5
```

e) Configure the file roller appender maximum file size by setting the value of the **log4j.appender.file.MaxFileSize** parameter.

Example

```
log4j.appender.file.MaxFileSize=100M
```

3. Configure the web server settings in the <IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/admin-ws.properties file.

   The web server is attached to AD Configuration Server.

   a) Configure the port that the web server listens on by setting the value of the **port** parameter.

   The default value is 8080.

   b) Configure the path to the configuration file of the logger by setting the value of the **log-conf-file** parameter.

   The default value is <IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/webservice.log4j.properties.

4. By default, the HTTP protocol is used to run the web service. To use the SSL/HTTPS protocol, follow the steps:

   a) Generate a self-signed key pair and store it in a Java keystore by using the Java Keytool command-line interface. Run the following command:

   Note: Java SDK or Java JRE must be installed, and the *JAVA_HOME* and *PATH* environment variables must be configured for the Java SDK or Java JRE.

```
keytool -genkeypair -keyalg RSA -alias {alias}
-ext SAN=DNS:localhost,IP:127.0.0.1 -dname {dname}
-validity {validity} -keysize 2048 -keypass {keypass}
-storepass {storepass} -keystore {keystore}
```

{alias}
The name that is used by the Java keystore to identify the generated key. The name must be unique within the Java keystore.

{dname}
The distinguished name from the X.500 standard. This name is associated with the alias for the key pair in the keystore. Also, the name is used as the value in the "issuer" and "subject" fields in the self-signed certificate.

{validity}
The number of days that the certificate that is attached to the key pair is valid.

{keypass}
The password that is needed to access the key pair within the keystore.

{storepass}
The password for the Java keystore.

{keystore}
The path to the keystore file, which is used to store the generated key pair. If the file does not exist, a keystore file is created.

Example

```
keytool -genkeypair -keyalg RSA -alias my-key-pair
-ext SAN=DNS:localhost,IP:127.0.0.1 -dname CN="IBM AD"
-validity 9999 -keysize 2048 -keypass my-key-password
-storepass my-store-password -keystore C:\my_keystore
```

b) Configure the web server that is attached to IBM AD Configuration Server to use the SSL/HTTPS protocol. In the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/keystore-config.properties` file, configure the following parameters:

`path`
Set the value to the path of the Java keystore that is generated in the preceding substep.

`storepass`
Set the value to the password for the Java keystore.

`keypass`
Set the value to the password that is needed to access the key pair within the keystore.

c) In the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/admin-ws.properties` file, set the value of the **`keystore-conf-file`** parameter to the path of the keystore configuration file.

5. Make sure that **IBM Application Discovery Configuration Service Admin** is up and running. If any error occurs, start the service manually:

- **On Windows**

  a. Go to **Start Menu** > **IBM Application Discovery and Delivery Intelligence** and **Launch IBM Application Discovery Configuration Service Admin**.

  b. If the service does not start, check the `server.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/` folder.

- **On Linux**

  a. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/` and locate the `startServer.sh` and `startWebServerUI.sh` files.

  b. Open a terminal under the current directory and run the following commands to launch the **IBM Application Discovery Configuration Service Admin**.

  ```
  startServer.sh
  startWebServerUI.sh
  ```

  c. In case that the `.sh` files are not executable, open a terminal and run the following commands for flagging them as executable:

  ```
  chmod +x startServer.sh
  ```

  and

  ```
  chmod +x startWebServerUI.sh
  ```

d. If the service does not start, check the `server.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/` folder.

Important: For monitoring the **IBM AD Configuration Server** tasks, see "Finding Log Files" on page 537.

STEP 2. Configuring IBM AD Web Services

IBM AD Catalog

The **IBM AD Catalog** package is used to enable the impact analysis on the APIs published by a **z/OS Connect** server.

Configuring the Catalog Service

Note: The Catalog Service uses only one database at a time and it can be created on **SQL Server** or **Db2 for LUW**.

Configuring the Catalog Service to run on SQL Server

By performing the following actions, the data is imported in the SQL database.

1. Open the `server.xml` file from `\wlp\usr\servers\ad_server`.
2. Enable **IBM AD Catalog**, by removing the `<!-- -->` comment around its `<webApplication>` element.

```
<webApplication contextRoot="com.ibm.etools.ad.catalog" id="com.ibm.etools.ad.catalog"
location="com.ibm.etools.ad.catalog.war" name="ad-catalog">
        <application-bnd>
            <security-role name="ADCatalogUser">
                <user name="catalog"/>
            </security-role>
        </application-bnd>
</webApplication>
```

3. Open the `server.xml` file from `\wlp\usr\servers\ad_server`.
4. In the *<!-- Add the Catalog database connection details-->* area, enter the SQL database in the **databaseName** field.

```
databaseName="Catalog"
```

5. In the same area, enter the IP of the server in the **serverName** field.

```
serverName=
```

6. Enter the SQL port in the **portNumber** field (the default is 1433).

```
portNumber="1433"
```

7. Enter the user and password used to connect to the SQL database in the **user** and **password** fields.

```
<!-- Add the Catalog database connection details-->
<dataSource id="ADCatRDB" jndiName="jdbc/ad/catalog/relational" type="javax.sql.DataSource">
    <jdbcDriver libraryRef="JTDSLib" javax.sql.DataSource="net.sourceforge.jtds.jdbcx.JtdsDataSource"/>
    <properties databaseName="catalog" serverName="127.0.0.1" portNumber="1433" user="sa" password="password"/>
</dataSource>
```

Important: The password that is used to connect to the SQL database can be encrypted. The **securityUtility** command supports plain text encryption for Liberty. For more information, see securityUtility command.

8. Make sure that **IBM® AD Catalog Service** is configured in **IBM® AD Configuration Server**. For more information, see " Configuring IBM AD Catalog Service" on page 47.
9. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

Note:

- Wait until the services are restarted, this can take a few minutes to complete.
- The Catalog database will be automatically populated when **Data Collector** starts for the first time.

Configuring the Catalog Service to run on Db2 for LUW

By performing the following actions, the data is imported in the Db2 for LUW (Linux, UNIX, and Windows) database.

1. Open the `server.xml` file from `\wlp\usr\servers\ad_server`.
2. Enable **IBM AD Catalog**, by removing the `<!-- -->` comment around its `<webApplication>` element.

```
<webApplication contextRoot="com.ibm.etools.ad.catalog" id="com.ibm.etools.ad.catalog"
location="com.ibm.etools.ad.catalog.war" name="ad-catalog">
        <application-bnd>
            <security-role name="ADCatalogUser">
                <user name="catalog"/>
            </security-role>
        </application-bnd>
</webApplication>
```

3. In the *<!-- Add the Catalog database connection details-->* area, add the following library.

```
<library id="DB2JCCLib">
        <fileset dir="${shared.config.dir}" includes="db2jcc4-4.21.29.jar
com.ibm.ez.db2jcc.license.cisuz-1.0.0.jar"/>
</library>
```

Important: Make sure that in the `wlp/usr/shared/config` folder, the two `.jar` files are present. Depending on the Db2 version that is installed on your machine, the `.jar` file names can differ.

4. In the *<!-- Add the Catalog database connection details-->* area, edit the `<dataSource>` XML element as follows.

Example:

```
<dataSource id="db2" jndiName="jdbc/ad/catalog/relational">
        <jdbcDriver libraryRef="DB2JCCLib"/>
        <properties.db2.jcc databaseName="CATDB" serverName="localhost"
portNumber="50000" user="db2admin" password="{aes}ABitSh1jv1T9i8tpGN7uQiSvx+yAZl+
+igoVWxzzMu3CoI4lM6d69HP69MuRDeIEDQ=="/>
</dataSource>
```

Note: When configuring Catalog Service to run on Db2 for LUW, make sure that `properties.db2.jcc` is present in the `<dataSource>` XML element.

Where:

- **databaseName** - expects the Db2 database name.
- **serverName** - expects the name/IP of the server when Db2 Server is installed.
- **portNumber** - expects the port number configured for the communication with Db2 Server. The default value is 50000.
- **user** - expects the user that is used to connect to the Db2 database.
- **password** - expects the password that is used to connect to the Db2 database.

  Important: The password that is used to connect to the Db2 database can be encrypted. The **securityUtility** command supports plain text encryption for Liberty. For more information, see securityUtility command.

5. Make sure that **IBM AD Catalog Service** is configured in **IBM AD Configuration Server**. For more information, see “ Configuring IBM AD Catalog Service” on page 47.
6. Once the configuration is done, go to the **Dashboard** tab, in **IBM AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

Note:

- Wait until the services are restarted, this can take a few minutes to complete.

- The Catalog database will be automatically populated when **Data Collector** starts for the first time.

Configuring the Data Collector

Follow below steps to configure **Data Collector** that gathers the information related to the services exposed by the mainframe.

1. Copy the files for **Data Collector** from `zoscDataCollector` to a new working directory.
2. Update the `DC.properties` file in this newly created working directory, by setting the following property values:

   Important: Please do not update other property values in the file, unless otherwise instructed by IBM Support.

   a. **ZoscURL**: Specify the protocol, the host name and the port for the **z/OS Connect** server as a URL.

      Note: You need to append `/zosConnect/apis` as the path part of the URL. An example value is *http://<zoscHost>:9081/zosConnect/apis*.

   b. **ZoscUser**: Specify the user name by which **Data Collector** connects to the **z/OS Connect** server. If user authentication is disabled on **z/OS Connect** server or **IBM AD Catalog**, you do not need to define this property value in the property file

   c. **ZoscPass**: Specify the password (in plain text) used by **Data Collector** to connect to the **z/OS Connect** server. If user authentication is disabled on **z/OS Connect** server or **IBM AD Catalog**, you do not need to define this property value in the property file

   d. **DataCollectorId**: Specify the **Data Collector** ID, for **AD** to identify an instance of **z/OS Connect** server.

      Note: For v5.0.4, the **DataCollectorId** string must be the same as the **ZoscURL** property value. Additionally, **DataCollectorId** must contain the *zosConnect/apis* string which can be either *zosConnect/apis* or the full URL of the *ZoscURL*.

   e. **ADCatalogURL**: Specify the protocol, the host name and the port used by the **AD Catalog**.

      Note: You need to append `com.ibm.etools.ad.catalog/cat/entity` to the URL path. An example value is *http://<catalogHost>:9080/com.ibm.etools.ad.catalog/cat/entity/*.

   f. **ADCatalogUser**: Specify the user name by which **Data Collector** connects to the **AD Catalog** If user authentication is disabled on **z/OS Connect** server or **IBM AD Catalog**, you do not need to define this property value in the property file.

   g. **ADCatalogPass**: Specify the password (in plain text) used by **Data Collector** to connect to the **IBM AD Catalog** If user authentication is disabled on **z/OS Connect** server or **AD Catalog**, you do not need to define this property value in the property file.
3. Check the PATH environment variable. You need to add the directory where `java.exe` resides in the *PATH environment* variable, in case it is missing.
4. Update `zoscdc.cmd`.

   Important: You need to specify the jar file of **JSON4J** that resides under `/WAS_Liberty_Library/lib/` with full path name, as a part of the *-classpath* argument. The file name of the jar file should look like `com.ibm.json4j_x.x.x.jar`.

   Alternatively, you can specify the property values listed in **step 2** as the property values passed to the java command. For example, the **ZoscUser** property can be specified as an argument for **-DZoscUser=user1**.

How to Collect z/OS Connect API Information

Take the following steps to invoke **Data Collector** to collect **z/OS Connect** API information.

1. Open a windows command prompt and change the current directory to the `/Data_Collector_Working_Directory/`.
2. Run `zoscdc.cmd`.

Configuring IBM AD Components to Use the Audit Service

IBM AD Build Client, **IBM AD Batch Server** and **IBM AD Analyze Client** can be configured to use the **Audit Service**.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **General** tab.
2. The **Audit Service** for **IBM AD Batch Server** is set to be enabled by default. To stop using the **Audit Service**, set **`audit.enable=true`** to **`audit.enable=false`** in the `server.properties` file located in the `\conf` folder.

```
##Audit server
# default true
#audit.enable=true
```

3. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

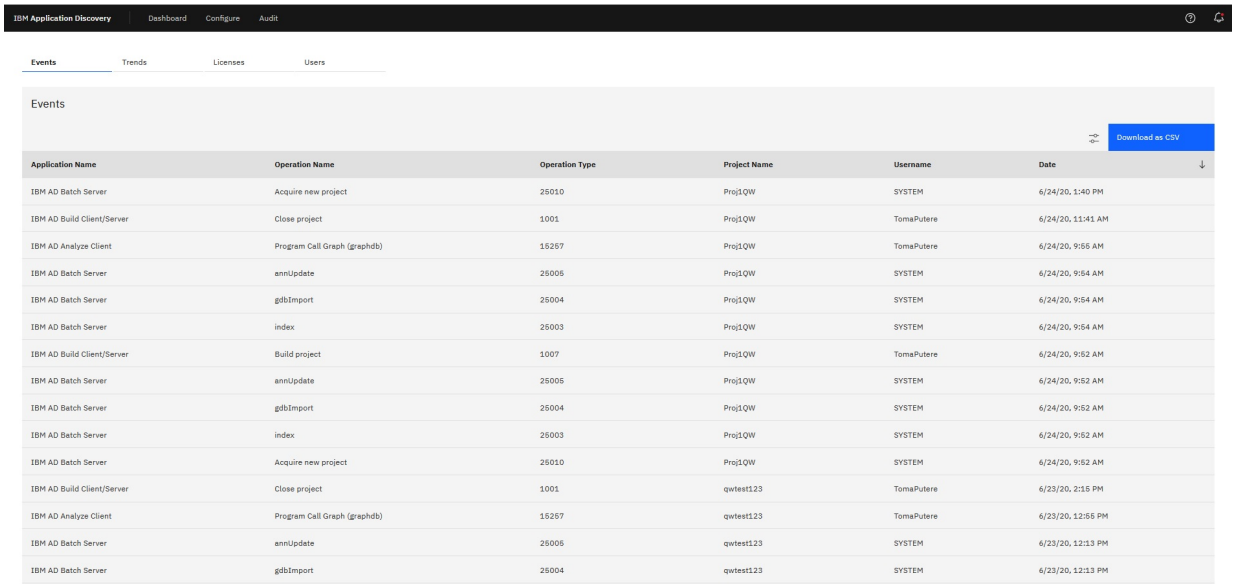
   Note:

   - Wait until the services are restarted, this can take a few minutes to complete.
   - The Audit database will be automatically populated when **IBM Application Discovery Web Service** starts.

Using the Audit Service Web GUI

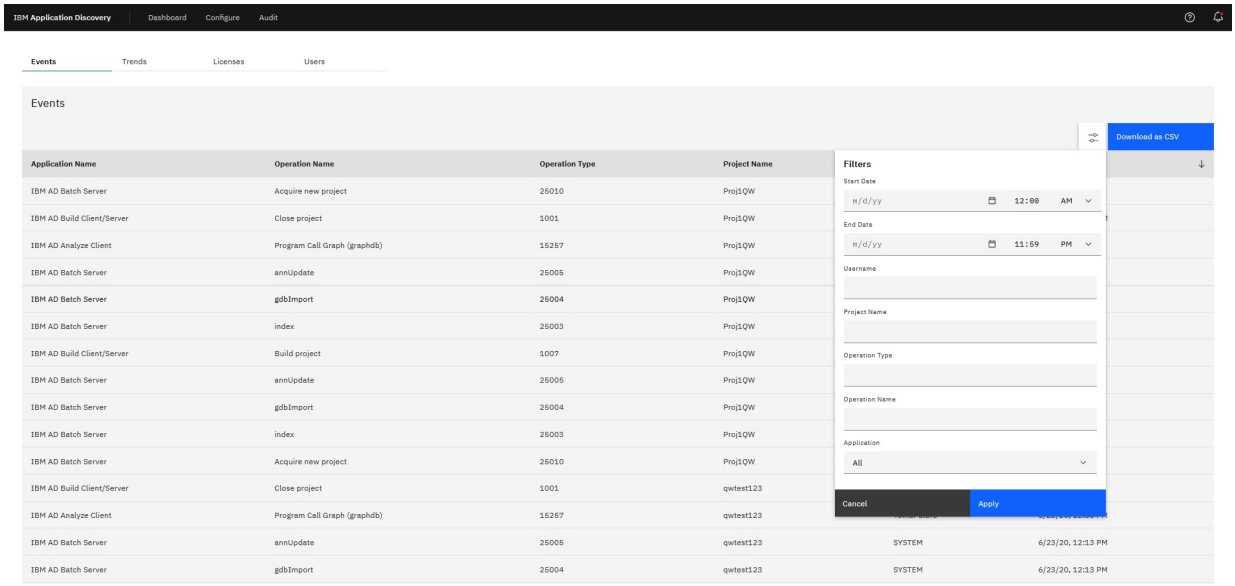
1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Audit** > **Events**.

   As a result, the **Audit** page is displayed as in the following image.



Application Name	Operation Name	Operation Type	Project Name	Username	Date
IBM AD Batch Server	Acquire new project	25010	Proj1QW	SYSTEM	6/24/20, 1:40 PM
IBM AD Build Client/Server	Close project	1001	Proj1QW	TomaPutere	6/24/20, 11:41 AM
IBM AD Analyze Client	Program Call Graph (graphdb)	15267	Proj1QW	TomaPutere	6/24/20, 9:55 AM
IBM AD Batch Server	annUpdate	25005	Proj1QW	SYSTEM	6/24/20, 9:54 AM
IBM AD Batch Server	gdbImport	25004	Proj1QW	SYSTEM	6/24/20, 9:54 AM
IBM AD Batch Server	index	25003	Proj1QW	SYSTEM	6/24/20, 9:54 AM
IBM AD Build Client/Server	Build project	1007	Proj1QW	TomaPutere	6/24/20, 9:52 AM
IBM AD Batch Server	annUpdate	25005	Proj1QW	SYSTEM	6/24/20, 9:52 AM
IBM AD Batch Server	gdbImport	25004	Proj1QW	SYSTEM	6/24/20, 9:52 AM
IBM AD Batch Server	index	25003	Proj1QW	SYSTEM	6/24/20, 9:52 AM
IBM AD Batch Server	Acquire new project	25010	Proj1QW	SYSTEM	6/24/20, 9:52 AM
IBM AD Build Client/Server	Close project	1001	qwtest123	TomaPutere	6/23/20, 2:15 PM
IBM AD Analyze Client	Program Call Graph (graphdb)	15267	qwtest123	TomaPutere	6/23/20, 12:55 PM
IBM AD Batch Server	annUpdate	25005	qwtest123	SYSTEM	6/23/20, 12:13 PM
IBM AD Batch Server	gdbImport	25004	qwtest123	SYSTEM	6/23/20, 12:13 PM

2. Use filters to search for specific entries by start and end date, username, project name, operation type, operation name, or by application.



3. Alternatively, you can export the results by clicking **Download as CSV**.

```
Application Name, Operation Name, Operation Type, Project Name, Username, Date, IP, Instance Id
IBM AD Build Client/Server,Open Project,1000,SQL_Within,Username,03.03.2020 at 03:00:36,9.228.129.107,null
IBM AD Build Client/Server,Build project,1007,SQL_Within,Username,03.03.2020 at 03:11:05,9.228.129.107,null
IBM AD Build Client/Server,Build project,1007,SQL_Within,Username,03.03.2020 at 03:43:01,9.228.129.107,null
IBM AD Build Client/Server,Open Project,1000,Defect24633DBDFile,Username,03.03.2020 at 03:43:12,9.228.129.107,null
IBM AD Build Client/Server,Close project,1001,SQL_Within,Username,03.03.2020 at 03:43:15,9.228.129.107,null
IBM AD Batch Server,gdbImport,25004,Defect24633DBDFile,SYSTEM,03.03.2020 at 03:57:57,9.228.129.107,4ed6d264-441e-4f8d-bb9d-c8c7302e41fc
IBM AD Batch Server,index,25003,Defect24633DBDFile,SYSTEM,03.03.2020 at 03:57:57,9.228.129.107,4ed6d264-441e-4f8d-bb9d-c8c7302e41fc
IBM AD Batch Server,annUpdate,25005,Defect24633DBDFile,SYSTEM,03.03.2020 at 03:57:57,9.228.129.107,4ed6d264-441e-4f8d-bb9d-c8c7302e41fc
```

Modifying the logging level

You can modify the logging level of the Audit Service by either updating the configuration file or sending a PUT request to the REST API. Accepted logging level values are INFO, WARN, and ERROR. If you modify the logging level at run time, it takes five seconds for the change to take effect.

1. To modify the logging level by updating the configuration file, open *<IBM ADDI Installation Folder>*`\IBM AD Web Services\wlp\usr\servers\ad_server\apps\expanded\ad-audit.war\WEB-INF\classes\log4j2.xml`, find the com.ibm.ad.audit logger, change the value of the `level` attribute, and then save the change.
2. To modify the logging level by sending a PUT request, use the *libertyURL*`/ad-audit/logging/change-level` endpoint, and send the request body with input.

   Note: This is a temporary change, which is lost after the Liberty server restart.

   Example

   ```
   http://localhost:9080/ad-audit/logging/change-level
   ```

   ```
   payload:
   {
       "name":"com.ibm.ad.audit",
       "level":"INFO"
   }
   ```

Modifying the wait time for receiving IBM AD Batch Server status

After an action is finished in IBM AD Build Client, audit events are expected to be sent from IBM AD Batch Server in a given time period. By default, the Audit Service waits for two minutes. If longer wait time is needed, you can follow the steps to modify the value.

1. Stop IBM Application Discovery WebSphere Liberty Profile Service.

   To stop the service, open Windows Services (`services.msc`), locate IBM Application Discovery WebSphere Liberty Profile Service, and then click **Stop**.

2. Open *<IBM ADDI Installation Folder>*\IBM AD Web Services\wlp\usr\servers\ad_server\conf.audit\audit.properties, add the following line, and then save the change.

```
com.ibm.ad.audit.timeout.batch=time_value
```

com.ibm.ad.audit.timeout.batch
Specifies the wait time for receiving IBM AD Batch Server status. Specify the value in milliseconds, for example, 120000, which equals two minutes.

3. Restart IBM Application Discovery WebSphere Liberty Profile Service.

   To restart the service, open Windows Services (services.msc), locate IBM Application Discovery WebSphere Liberty Profile Service, and then click **Start**.

IBM AD BRD

IBM AD BRD allows IBM AD to save data that is used to define Business Rules within IBM ADDI Extension.

Configuring the BRD Service

1. Open the server.xml file from \wlp\usr\servers\ad_server.
2. Enable **IBM AD BRD**, by removing the <!-- --> comment around its <webApplication> element.

```
<webApplication contextRoot="brd-ws" id="com.ibm.ad.brd.restapi"
location="com.ibm.ad.brd.restapi.war" name="ad-brd">
        <classloader privateLibraryRef="config.brd-ws"/>
    </webApplication>
```

3. Go to \wlp\usr\servers\ad_server\conf.brd-ws and in application.properties file, enter the desired values for the properties detailed below.
4. Define the basic authentication username and password for IBM AD BRD Web Service. Once configured, these credentials will be used to create a server connection to IBM AD BRD Server in ADDI Extension.

   Note: The username is case-insensitive while the password is case-sensitive.

```
## Define the basic authentication username for IBM AD BRD Service
## The username is case-insensitive
ad.user=

## Define the basic authentication password for IBM AD BRD Service
## The password is case-sensitive
ad.password=
```

5. Add the IP of the computer where the **IBM AD Configuration Server** host is installed.

```
## IBM AD Configuration Server host
ccs.server.host=
```

6. Add the port number to be used by **IBM AD Configuration Server**; the default port is 2181, if no other value is specified the default value will be used.

```
## IBM AD Configuration Server port
## default 2181
#ccs.server.port=
```

7. Add the ID of the Environment set in **IBM AD Configuration Server**.

```
## IBM AD Configuration Server environment
ccs.environment=
```

8. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

   Note: Wait until the services are restarted, this can take a few minutes to complete.

STEP 3. (Optional) Configuring IBM AD Validation Service

IBM AD Validation Service component is specific only for ChangeMan ZMF users, therefore it is not part of the *must have* components installation.

IBM AD Validation Service is automatically installed during the **IBM AD Build** installation.

IBM AD Validation Service acts like a listener and is linked directly with **IBM AD Connect for Mainframe** component (Mainframe Agents).

After **IBM AD Validation Service** is installed, go to `<IBM AD Build Client installation folder>\Bin\Release\IBMApplicationDiscoveryValidationServer\SampleConf`.

Select all configuration files and copy them to `<IBM AD Build Client installation folder>\Bin\Release\IBMApplicationDiscoveryValidationServer.`

Next, perform the following configurations.

1. Configure `ProjectsMapping.txt` to have a valid input. This is the configuration file for defining the mapping between the projects that are used to download mainframe members, applications, and subsystems.

   Note: Comparing with the `ProjectsMappingParallelBuild.txt` file, the projects that are specified in the `ProjectsMapping.txt` file do not need to contain the virtual folder that is specified in the `FoldersMapping.txt` file, as they are not used for builds. However, a z/OS connection must be attached and configured to the projects.

   Each line of the configuration file must have the following comma-separated values format:

   `<ProjectName>, <ApplicationName>, <Subsystem>`

   Note:

   - `<ProjectName>` represents the project that is defined in **IBM AD Build Client**.
   - `<ApplicationName>` and `<Subsystem>` are defined in **ChangeMan ZMF**.

   Example of the configuration file:

   ```
   Project1, App1, Subsys1
   ```

2. Configure `IncludesOrder.txt` to have a valid input. This is the configuration file for defining the ChangeMan Baseline Libraries Types and the order of COBOL Includes locations. This configuration file is used later on while you set up the path for the COBOL Include folders

   The configuration file must have the following comma-separated values format:

   `<Type1>, <Type2>,…, <Type`*`n`*`>`

   Example of the configuration file:

   ```
   CPY, INC, IND, CPA
   ```

   Note: It is EXTREMELY important to add the types in the order in which the include files must be looked after.

3. Configure `FoldersMapping.txt` to have a valid input. This is the configuration file for defining a mapping between a type of a mainframe member, that is defined in ChangeMan ZMF, and a virtual folder name of an **IBM AD** project. This configuration file is used during the synchronize phase of the validation process.

   Each line of the configuration file must have the following comma-separated values format:

   `<MemberType>, <VirtualFolderName>`

   - `<MemberType>` is defined in **ChangeMan ZMF**.
   - `<VirtualFolderName>` is defined in **IBM AD Build Client**.

   Examples of the configuration file:

```
COB, zOS Cobol
```

```
ASM, Assembler
```

4. Configure `ServicePort.txt` to have a valid input. This is the configuration file for defining the Service's port.

   The configuration file must have the following format:

   `<Port Number>`

   Any available port can be used, for example:

   ```
   48000
   ```

5. Enable or disable sending feedback to the mainframe by configuring the `LoopbackResults.txt` file with one the following values:

   Y
   Enables sending feedback to the mainframe according to the weight of rules.

   N
   Disables sending feedback to the mainframe.

6. Set parallel validation parameters for the maximum-allowed values by configuring the `ParallelValidationParameters.txt`.

   The configuration file must have the following comma-separated values format:

   `<Number_of_validations_in_parallel>,<Number_of_components_per_validation>`

   Note:

   - Do not set the number of validations in parallel greater than the number of CPU cores. Otherwise, the validation process might be unstable.
   - Do not set the number of components per validation greater than 20. Otherwise, the performance might be negatively affected.

   Examples of the configuration file:

   ```
   4,10
   ```

   Allows a maximum of four validation instances in parallel, and a maximum of 10 stages or members that are allocated for each instance. You can set these values for a computer with 4 CPU cores.

   ```
   8,15
   ```

   Allows a maximum of eight validation instances in parallel, and a maximum of 15 members that are allocated for each instance. You can set these values for a computer with 8 CPU cores.

7. Configure the mapping between the projects that are used to compile the members to be validated in parallel, applications, and subsystems. Set the mapping values in the `ProjectsMappingParallelBuild.txt` file.

   Each line of the configuration file must have the following comma-separated values format:

   `<ProjectName>, <ApplicationName>, <Subsystem>`

   Note:

   - `<ProjectName>` represents the project that is defined in **IBM AD Build Client**.
   - `<ApplicationName>` and `<Subsystem>` are defined in **ChangeMan ZMF**.
   - The number of the projects that are mapped to one pair of an application and a subsystem must be greater than or equal to the maximum number of validations in parallel, which is specified in the `ParallelValidationParameters.txt` file. Otherwise, the service cannot start.
   - Comparing with the `ProjectsMapping.txt` file, the projects that are specified in the `ProjectsMappingParallelBuild.txt` file do not need to have a z/OS connection that is

attached and configured, as they are used only for builds. However, the projects must contain the virtual folder that is specified in the `FoldersMapping.txt` file.

The following example shows the mapping configurations for eight validations in parallel:

```
Project1, App1, Subsys1
Project2, App1, Subsys1
Project3, App1, Subsys1
Project4, App1, Subsys1
Project5, App1, Subsys1
Project6, App1, Subsys1
Project7, App1, Subsys1
Project8, App1, Subsys1
Project9, App2, Subsys1
Project10, App2, Subsys1
Project11, App2, Subsys1
Project12, App2, Subsys1
Project13, App2, Subsys1
Project14, App2, Subsys1
Project15, App2, Subsys1
Project16, App2, Subsys1
```

8. Configure the completion code for messages by configuring the `CompletionCodeVsMessage.txt` file.

   Each line of the configuration file must have the following pipe-delimited format:

   `<CompletionCode>|<DescriptiveMessage>`

   Note: The descriptive message must have a maximum length of 23 **characters**.

   Example of the configuration file:

   ```
   0|Validation Success
   4|Validation Warning
   8|Validation Failed
   ```

   Each of the numbers in the example reflects the weight of the rule that is specified in the `ruleBased.properties` file.
9. Configure the approval request parameters in the `ApprovalRequestParameters.txt` file.

   Each line of the configuration file must have the following comma-separated values format:

   `<ProjectName>,<ProcLibrary>`

   Note:

   - `<ProjectName>` represents the project that is defined in **IBM AD Build Client**.
   - `<ProcLibrary>` is a PDS/E library that is defined in **ChangeMan ZMF**.
   - `<ProcLibrary>` must have a maximum length of 23 **characters**.

   Example of the configuration file:

   ```
   Project1,PJ.PROCLIB.S814
   ```

10. Configure the encryption policy by specifying a flag value in the `TLSEncryption.text` file.

    This configuration file contains a single flag byte. The flag can take the value Y or N. It is not case-sensitive. Y indicates that the Application Transparent Transport Layer Security (AT-TLS) encryption is enabled. If N is specified or the flag value is missing, the communication will not be encrypted.

1. Go to `<IBM AD Build Client installation folder\Bin\Release\IBMApplicationDiscoveryValidationServer\ReportsGenerator\sample-conf` folder. This `sample-conf` folder contains the templates for all the configuration files needed to customize the functioning of **ReportsGenerator**. To configure **ReportsGenerator**, copy the templates from the `sample-conf` folder, and place them in the `conf` folder.

   Once this step is completed and before you start **IBM AD Validation Service**, some preliminary configurations are required. You need to specify on which projects you want **ReportsGenerator** to run the reports, which reports to generate, where to store the generated reports etc. The configuration

parameters are stored in `server.properties`, `project.properties`, `ruleBased.properties` and `ruleBasedConfig.properties` files.

For more information, see the following sections that contain a detailed description of the parameters available in these four files:

- “Server Properties” on page 78
- “Global Settings” on page 81
- “ruleBased.properties File” on page 87
- “ruleBasedConfig.properties File” on page 87

Note: ReportsGenerator is invoked by Validation Server after the Build process.

2. Start **IBM AD Validation Service**: Click **Start** and then select **All Programs** > **IBM Application Discovery Build Client** > **Start IBM Application Discovery Validation Service**.

   The service can also be started from Windows Services (`services.msc`) by locating `IBMApplicationDiscoveryValidationServer` and pressing **Start**.

   Important: For monitoring the **Validation Service** tasks, see “Finding Log Files” on page 537.

STEP 4. Configuring IBM AD File Service

Follow the configuration steps that are needed to have up and running **IBM AD File Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD File Service available in IBM AD Configuration Server
3. Restart IBM AD File Service

1. Configure the parameters that are present in the `conf.yaml` file

Important: The configuration of the following parameters is not mandatory. It is recommended to follow below steps only in case that you had previously configured these parameters and you performed an upgrade to the latest version of **IBM AD** product.

On the machine where **IBM AD File Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

   ```
   #if true, tls information (key, cert) must be specified
   https: false
   ```

   b. If the **https** parameter is set to *true*, a secured communication is used.

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
#if true, tls information (key, cert) must be specified
https: true
```

2. If the **https** parameter is set to *true* and the TLS certificate for **IBM AD File Service** are generated, enter the paths of the `certificate(.crt)` and the `key(.key)` files. If the **https** parameter is set to *false* leave blank the following lines. Example:

```
#mandatory if https: true
tls:
    key: C:\certs\file.service.key
    cert: C:\certs\file.service.crt
```

3. Leave blank the line where the **authSrv** parameter is present since **Authentication Server (DEX)** is not needed.

```
#authentication server URL
authSrv:
```

4. The **caseSensitive** parameter can be set to *true* or *false*. Through this parameter, you set the mapping type (case-sensitive) of the folders under which the resources are located.

```
caseSensitive: false
```

5. The default value of the **disableAuth** parameter is *true*. Leave the default value since **Authentication Server (DEX)** is not needed.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: true
```

6. The default value of the **matchUsers** parameter is *false*, which means that the matching is made by **group names**. If you want to authorize users and not groups, set the **matchUsers** to true and define users in **IBM AD Configuration Server**. For more information, see “Adding a User” on page 49.

```
matchUsers: false
```

2. Make IBM AD File Service available in IBM AD Configuration Server

The access rights of users or users' groups are mapped to a certain folder that contains the source files. A user can start the analysis on the source files if it has read access rights.

After **IBM AD File Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD File Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **File Service**. The **File Service** settings page is displayed.
2. In the **Mappings** section, click the **Add** button to add a new mapping and enter the following information:
   - **Remote** - add the remote path that can be used to query this service, for example, a UNC path or a local path where the resources/projects reside.
   - **Local** - add the local path that mirrors the remote path. If missing, it is identical to **Remote**.

     Note:

     – In case you have multiple physical folders that reside resources or projects, an entry for each folder needs to be added in the **mapping** section.

     – In order to see **Flow Chart** analysis in **IBM AD Analyze Client** for a specific project, you need to add the path of the project folder or the path of the folder that contains all projects.

Examples:

- `\\9.20.128.222\Projects` - path to all projects.
- `\\9.20.128.222\ADProject` - path to a specific project.

- Define which **User Groups** can access the related mapping folders.

  Note: If no **User Groups** are defined, all users will have access to the related mapping folders.
- Click **Save**.

3. In the **Group Settings** section, enter the following information:
   - **Session Timeout** - add the session timeout to determine how long the service needs to remain connected to **IBM AD Configuration Server** before it must perform a new connection. The value format must be a whole number followed by 's' for seconds, 'm' for minutes or 'h' for hours.
   - **Groups Polling** - add the refresh time to check periodically the groups that are present in **IBM AD Configuration Server**. The value format must be a whole number followed by 's' for seconds, 'm' for minutes or 'h' for hours.
4. Click **Save**.

3. Restart IBM AD File Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **File Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. If the service does not start, check the `.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/` folder and locate the `micro-srcd-x.x.x.bin` file.
  2. In case that the `.bin` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x micro-srcd-x.x.x.bin
     ```
  3. If the service does not start, check the `.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/` folder.

STEP 5. Configuring IBM AD Manual Resolutions Service

Follow the configuration steps that are needed to have up and running **IBM AD Manual Resolutions Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Manual Resolutions Service available in IBM AD Configuration Server
3. Restart IBM AD Manual Resolutions Service

1. Configure the parameters that are present in the `conf.yaml` file

Important: The configuration of the following parameters is not mandatory. It is recommended to follow below steps only in case that you had previously configured these parameters and you performed an upgrade to the latest version of **IBM AD** product.

On the machine where **IBM AD Manual Resolutions Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Manual`

Resolutions Service/sample-conf/ and copy the conf.yaml file in the /conf folder. Open the conf.yaml file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Add the port of **IBM AD Configuration Server**.

   ```
   ## Coordination and Configuration Server port
   ## default 2181
   ccs.server.port: 2181
   ```

2. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

      ```
      #if communication should be secured with TLS
      https: false
      ```

   b. If the **https** parameter is set to *true*, a secured communication is used.

      Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (.crt) and a private key for the certificate (.key).

      ```
      #if communication should be secured with TLS
      https: true
      ```

   Note: If the **https** parameter is set to *true*, an additional step needs to be performed. Locate startServer.bat file under <IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/ and replace the following line:

   ```
   set tlsoptions=
   ```

   with:

   ```
   SET keystorepath=<"path_to_keystore">
   SET keystorepass=<"password_of_keystore">
   set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
   -Djavax.net.ssl.keyStorePassword="%keystorepass%"
   ```

   Where:

   - **Path to keystore** is the path to the keystore that holds the certificate for **IBM Manual Resolutions Service**.
   - **Keystore password** is the keystore password.

3. Leave blank the line where the **authSrv** parameter is present since **Authentication Server (DEX)** is not needed.

   ```
   #authentication server URL
   authSrv:
   ```

4. The default value of the **disableAuth** parameter is *true*. Leave the default value since **Authentication Server (DEX)** is not needed.

   ```
   #disable authentication/authorization. allow all files to be sent
   disableAuth: true
   ```

2. Make IBM AD Manual Resolutions Service available in IBM AD Configuration Server

The **IBM AD Manual Resolutions Service** manages the manually added resolutions and it is mandatory to be configured if you want to use Callgraph-based analyzes (graphs or reports).

After **IBM AD Manual Resolutions Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Manual Resolutions Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Manual Resolutions Service**. The **Manual Resolutions Service** settings page is displayed.
2. If necessary, you can modify the path where the **journal files** are created. The path where these files are generated is separated from the project's path and needs to be accessible only for **IBM AD Manual Resolutions Service**. Once a project is imported, a folder with the same name is generated in the related path and it hosts all the files that are needed to manage dynamic call resolutions. Example:

   ```
   C:\AD\Resolutions
   ```

3. Optionally, the main path where the manual resolutions are created for each project, can be overwritten. Click **Add**, select the target project, and add the path where the manual resolutions are created for the target project.
4. Click **Save**.

3. Restart IBM AD Manual Resolutions Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Manual Resolutions Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. If the service does not start, check the `manualres.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/log` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/` and locate the `startServer.sh` file.
  2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x startServer.sh
     ```

  3. If the service does not start, check the `manualres.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/log` folder.

STEP 6. Configuring IBM AD Mainframe Projects Service

Follow the configuration steps that are needed to have up and running **IBM AD Mainframe Projects Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Mainframe Projects Service available in IBM AD Configuration Server
3. Restart IBM AD Mainframe Projects Service

1. Configure the parameters that are present in the `conf.yaml` file

Important: The configuration of the following parameters is not mandatory. It is recommended to follow below steps only in case that you had previously configured these parameters and you performed an upgrade to the latest version of **IBM AD** product.

On the machine where **IBM AD Mainframe Projects Service** is installed, go to <IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/conf/ and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to <IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/sample-conf/ and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Add the port of **IBM AD Configuration Server**.

   ```
   ## Coordination and Configuration Server port
   ## default 2181
   ccs.server.port: 2181
   ```

2. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

   ```
   #if communication should be secured with TLS
   https: false
   ```

   b. If the **https** parameter is set to *true*, a secured communication is used.

   Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

   ```
   #if communication should be secured with TLS
   https: true
   ```

   Note: If the **https** parameter is set to *true*, an additional step needs to be performed.
   Locate `startServer.bat` file under <IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/ and replace the following line:

   ```
   set tlsoptions=
   ```

   with:

   ```
   SET keystorepath=<"path_to_keystore">
   SET keystorepass=<"password_of_keystore">
   set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
   -Djavax.net.ssl.keyStorePassword="%keystorepass%"
   ```

   Where:

   - **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Mainframe Projects Service**.
   - **Keystore password** is the keystore password.

3. The default value of the **disableAuth** parameter is *true*. Leave the default value since **Authentication Server (DEX)** is not needed.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: true
```

4. Leave blank the line where the **authSrv** parameter is present since **Authentication Server (DEX)** is not needed.

```
#authentication server URL
authSrv:
```

2. Make IBM AD Mainframe Projects Service available in IBM AD Configuration Server

Through **IBM AD Mainframe Projects Service** a user is authorized to access the AD projects.

After **IBM AD Mainframe Projects Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Mainframe Projects Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Mainframe Projects Service**. The **IBM AD Mainframe Projects Service** settings page is displayed.
2. Set the **Poll interval** parameter to determine how frequently the projects are tested (seconds).
3. Click **Save**.

3. Restart IBM AD Mainframe Projects Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Mainframe Projects Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. If the service does not start, check the `mfprojs.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/log` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/` and locate the `startServer.sh` file.
  2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x startServer.sh
     ```
  3. If the service does not start, check the `mfprojs.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/log` folder.

STEP 7. Configuring IBM AD Search Service

Follow the configuration steps that are needed to have up and running **IBM AD Search Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Search Service available in IBM AD Configuration Server

   Note: IBM AD Search Service is necessary for the Search in Files functionality to work. It is mandatory to have **IBM AD File Service** installed, configured and up and running.
3. Restart IBM AD Search Service

1. Configure the parameters that are present in the `conf.yaml` file

Important: The configuration of the following parameters is not mandatory. It is recommended to follow below steps only in case that you had previously configured these parameters and you performed an upgrade to the latest version of **IBM AD** product.

On the machine where **IBM AD Search Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

   ```
   #if communication should be secured with TLS
   https: false
   ```

   b. If the **https** parameter is set to *true*, a secured communication is used.

   Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

   ```
   #if communication should be secured with TLS
   https: true
   ```

   Note: If the **https** parameter is set to *true*, an additional step needs to be performed.
   Locate `startServer.bat` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/` and replace the following line:

   ```
   set tlsoptions=
   ```

   with:

   ```
   SET keystorepath=<"path_to_keystore">
   SET keystorepass=<"password_of_keystore">
   set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
   -Djavax.net.ssl.keyStorePassword="%keystorepass%"
   ```

   Where:

   - **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Search Service**.
   - **Keystore password** is the keystore password.

2. Leave blank the line where the **authSrv** parameter is present since **Authentication Server (DEX)** is not needed.

   ```
   #authentication server URL
   authSrv:
   ```

3. The default value of the **disableAuth** parameter is *true*. Leave the default value since **Authentication Server (DEX)** is not needed.

   ```
   #disable authentication/authorization. allow all files to be sent
   disableAuth: true
   ```

4. Optionally, the main path where the indexes are created for each project, can be overwritten by the following configuration. This path needs to be identical to the one present under `<IBM ADDI Installation Folder>/IBM Application Discovery Batch Server/conf/project.properties`, where the **project.projectName.index.indexFolder** parameter is present.

```
#optional
#overrides indexPath for a specific project.
projects:
   - project: Project1
     path: C:\Index\Indexes\Project1
```

2. Make IBM AD Search Service available in IBM AD Configuration Server

IBM AD Search Service is responsible with the access to the indexed data. The folder path in which the indexes are generated needs to be accessible for both **IBM AD Batch Server** and **IBM AD Search Service**.

After **IBM AD Search Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Search Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Search Service**. The **Search Service** settings page is displayed.
2. Add the path where the **indexes** are created. This path needs to have as an endpoint the same folder where the **indexes** are stored. Example:

```
\\server01\Indexes
```

3. Optionally, the main path where the indexes are created for each project, can be overwritten. Click **Add**, select the target project, and add the path where the indexes are created for the target project.
4. Click **Save**.

3. Restart IBM AD Search Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Search Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. Click **Start**, select **Run**, type **`services.msc`** and start **IBM Application Discovery Search Service**.
  3. If the service does not start, check the `search.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/log` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/` and locate the `startServer.sh` file.
  2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x startServer.sh
     ```

  3. If the service does not start, check the `search.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/log` folder.

STEP 8. Configuring IBM AD Batch Server

Before running **IBM AD Batch Server**, some preliminary configurations must be performed. You need to specify on which projects you want **IBM AD Batch Server** to run the reports, which reports to generate, where to store the generated reports, and so on. Also, you need to specify the parameters for **IBM AD Web Service**.

The configuration parameters are stored in `server.properties` and `project.properties` files, which can be found in the configuration folder.

Below are the instructions on how to perform a minimal configuration in order to have source code analysis in **IBM AD Analyze Client**. For detailed instructions on how to configure **IBM AD Batch Server**, see `IBM AD Batch Server User Guide`.

Note: Under Linux, in case `.sh` files are not executable, navigate to their installation directory, open a terminal and run the following command for flagging them as executable:

```
chmod +x filename.sh
```

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers** > **Graph Database Server**. The **Graph Database Server** settings page is displayed.
   a. During the installation process through **IBM AD Configuration Wizard**, the password for GraphDB Server's root user was set. If necessary, set a new password and confirm it.
   b. Click **Save**.
2. In the context of **Graph Database**, **OrientDB SSL** can be configured as follows:

   Note: For more information about OrientDB SSL configuration, see Using SSL with OrientDB.

   a. Copy the path to the keystore file that was uploaded to the AD Configuration Service Admin, for example, `<IBM ADDI Installation Folder>\security\<environment-id>\server_keystore.jks`.
   b. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\orientdb\orientdb-community\config`, open the `orientdb-server-config.xml`, set **OrientDB SSL** to `true` and update the password for both keystore and truststore as follows:

   ```
   <parameter name="network.ssl.clientAuth" value="true"/>
   <parameter name="network.ssl.keyStore" value="<IBM ADDI Installation
   Folder>\security\<environment-id>\server_keystore.jks"/>
   <parameter name="network.ssl.keyStorePassword" value="password"/>
   <parameter name="network.ssl.trustStore" value="<IBM ADDI Installation
   Folder>\security\<environment-id>\server_keystore.jks"/>
   <parameter name="network.ssl.trustStorePassword" value="password"/>
   ```

   Note: You can enter an encrypted **Keystore Password**. For more information, see "Obtaining an encrypted keystore password" on page 209.

   c. In the `listeners` section of the `orientdb-server-config.xml` file, update the first line that includes the socket and the default port range for **OrientDB SSL** as follows:

   ```
   <listener protocol="binary" socket="ssl" port-range="2424" ip-address="0.0.0.0"/>
   ```

   d. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\orientdb\orientdb-community\bin`, open `shutdown.bat`, find the line that starts with `set JAVA_OPTS`, and update the password for both keystore and truststore as follows:

   ```
   set JAVA_OPTS=-Djava.awt.headless=true -Dclient.ssl.enabled=true
   -Djavax.net.ssl.keyStore="<IBM ADDI Installation Folder>\security\<environment-
   id>\server_keystore.jks" -Djavax.net.ssl.keyStorePassword=password
   -Djavax.net.ssl.trustStore="<IBM ADDI Installation Folder>\security\<environment-
   id>\server_keystore.jks" -Djavax.net.ssl.trustStorePassword=password
   ```

   Note: Enter the same password that was entered in step "2.b" on page 76.

   e. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\`, open `startServer.bat` and set the **gdbOptions** parameter as follows:

   ```
   set gdbOptions=-DgdbImport.client.ssl.enabled=true
   -DgdbImport.javax.net.ssl.trustStore=<<IBM ADDI
   Installation Folder>\security\<environment-id>\server_keystore.jks>
   ```

```
-DgdbImport.javax.net.ssl.trustStorePassword=password
-DgdbImport.javax.net.ssl.keyStore=<<IBM ADDI Installation Folder>\security\<environment-
id>\server_keystore.jks> -DgdbImport.javax.net.ssl.keyStorePassword=password
```

Note: At this point, **OrientDB SSL** is configured to run with the **root** username and the password that is configured above.

3. The `project.properties` file contains a set of global settings, followed by the specific settings for each type of component. The global settings specify the projects on which the **IBM AD Batch Server** will operate and which components will run on the specified projects.

   Detailed information on the additional components that can be considered for this service.

Table 4. Optional Components

Component	Description
ruleBased	The **Rule Based** component generates reports for the resources specified in the configuration files according to the rules and parameters defined in the corresponding configuration files. **Note:** If the **Rule Based** component is used, make sure that the `ruleBased.properties` file is configured. For more information, go to "ruleBased.properties File" on page 87.
reports	The **Reports** component is used to generate the complexity reports. For more information, go to "Complexity Reports" on page 359.
cobolPP	The **cobolPP**, **jclPP**, and **pl1PP** components generate the expanded sources for Cobol, JCL, and PL/I. For more information, go to "View Expanded Source" on page 318.
jclPP	
pl1PP	
wsmetrics	The **wsmetrics** component is needed only if IBM ADDI Extension is used on the system. **Note:** Additionally, in order for the **wsmetrics** component to be executed, make sure that the **gdbImport** component is included in the components list of this service.
adidx	The **addix** component is needed only if IBM ADDI Extension is used on the system. The ADI Index component will index the resources of a project so that the Business Rules Discovery (BRD) feature can display code snippets.

4. On **Linux** only, Open the `mount.properties` file, located under `<installation folder>\IBM Application Discovery Batch Server/conf` folder and specify how the windows shared folders are mounted on the local files system, using the following pattern:

```
\\\\machine IP\\WindowsSharedFolder=/home/user/LinuxFolder
```

   Example:

```
\\\\192.168.56.57\\ProjectsSharedPathWindows=/home/user/ ProjectsSharedPathLinux
```

5. Optional step: for integration with IBM ADDI Extension only, please follow this procedure to set up the **AD Batch Server Web Service**:

   a. Go to *<IBM ADDI Installation Folder>*`\IBM Application Discovery Batch Server\`, and run `authConfigTool.bat` on **Windows** or `authConfigTool.sh` on **Linux**. A command prompt dialog window is displayed. Follow the directions and enter the username and the password that are used by the Web Service then press ENTER. `AuthConfigTool.bat` sets the user and password for Web Service basic access authentication.

b. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\conf folder, locate the webService.properties file, and set the **keystore-file** parameter to enable an encrypted communication. Example:

```
## ssl keystore file
keystore-file=keystore.jks
```

Note: The keystore file needs to be added in the same \conf folder where webService.properties is present.

- Restart **Graph Database Server**.
  - Under Windows: Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Graph Database Server**, and select **Restart Service**.

    Note: Wait until the service is restarted, this can take a few minutes to complete.
  - Under Linux: Go to *<IBM ADDI Installation Folder>*\IBM Application Discovery Batch Server\orientdb\orientdb-community-2.1.25-ibm1\bin\ and run server.sh. Make sure this process remains alive.
- Restart **Batch Server Service**.
  - Under Windows: Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Batch Server Service**, and select **Restart Service**.

    Note: Wait until the service is restarted, this can take a few minutes to complete.
  - Under Linux: Go to *<IBM ADDI Installation Folder>*\IBM Application Discovery Batch Server\, and run StartServer.sh. Make sure this process remains alive.
- (Only in case step 5 from above has been taken) Start **IBM AD Web Service**.
  - Under Windows: Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

    Note: Wait until the service is restarted, this can take a few minutes to complete.
  - Under Linux: Go to *<IBM ADDI Installation Folder>*\IBM Application Discovery Batch Server\, and run startWBServer.sh. Make sure this process remains alive.

Note: Make sure to restart **IBM AD Batch Server** after modifying the configuration files.

Important: For monitoring the **IBM AD Batch Server** tasks, see "Finding Log Files" on page 537.

Step 8.1 Advanced Configurations for IBM AD Batch Server

After **IBM Application Discovery for IBM Z (AD) Batch Server** is installed, a folder named sample-conf is created in the **IBM AD Batch Server** installation folder. This sample-conf folder contains the templates for all the configuration files needed to customize the functioning of **IBM AD Batch Server**. To configure **IBM AD Batch Server**, copy the templates from the sample-conf folder, and place them in the conf folder.

Once the setup is completed, before running **IBM AD Batch Server**, some preliminary configurations should be performed. You need to specify on which projects you want **IBM AD Batch Server** to run the reports, which reports to generate, where to store the generated reports etc. Also, you need to specify the parameters for **IBM AD Web Service**.

The configuration parameters are stored in server.properties, project.properties, and webservice.properties files. The following sections contain a detailed description of the parameters available in these three files as well as a description of other configuration files for specific components such as **Annotations**, **GraphDB**, **Search Index**, **Rule Based**, **Reports**, and **Web Service Metrics**.

Important: It is vital that you set the parameters for the **Search Index**, **Annotations** and **GraphDB** components. Failing to do so will make some of the analyses impossible to generate in **IBM AD Analyze**.

Server Properties

In server.properties file, enter the desired values for the properties that are detailed below.

Note: Make sure to restart **IBM AD Batch Server** after you modify the configuration file.

1. Add the IP of the computer where the **IBM AD Configuration Server** host is installed.

```
## Coordination and Configuration Server host
ccs.server.host=127.0.0.1
```

2. Add the port number to be used by **IBM AD Configuration Server**; the default port is 2181, if no other value is specified the default value is used.

```
## Coordination and Configuration Server port
## default 2181
#ccs.server.port=2181
```

3. Add the ID of the Environment set in **IBM AD Configuration Server**.

```
## Coordination and Configuration Server environment
ccs.environment=fb9107c2-8770-43ef-990c-15b387f19698
```

4. By default the audit of **IBM AD Batch Server** components is enabled, by setting it to false the Audit stops.

```
##Audit server
# default true
#audit.enable=true
```

5. Add the IP used by **IBM AD Batch Server** to communicate with **IBM AD Configuration Server**

```
## client host
## default: one of the available ip-s
# client.host=
```

6. Add the IP of the computer where **IBM AD Batch Server** is installed.

```
##aplication server host
#application.server.host=
```

7. Add the port number where **IBM AD Batch Server** listens to admin requests; the default port is 8686, if no other value is specified the default value is used.

```
## port on which the server listens to admin requests
## default: 8686
#application.server.port=
```

8. Specify how much to wait before the server stops (milliseconds); the default value is 5000, if no other value is specified the default value is used.

```
## how much to wait for the server to stop (milliseconds)
## default: 5000
#stop.timeout=
```

9. Specify how frequently database version changes are tested (seconds); the default value is 120, if no other value is specified the default value is used.

```
## how frequently database version changes are tested (seconds)
## default: 120
#poll.interval=
```

10. Specify the installation directory or the place where **IBM AD Batch Server** is installed. All paths are either absolute or relative to the installation directory. Relative paths start with the first directory under `${install.dir}`.

```
## ${install.dir} place where server is installed
## all paths are either absolute or relative to ${install.dir}
## relative paths start with the first directory under ${install.dir}
```

    a) Enter the path to the directory where configuration files are stored; the default path is `${install.dir}/conf`; if no other path is specified the default path is used.

```
## path to the directory where configuration files are placed
## default: ${install.dir}/conf
conf.dir={install.dir}/conf
```

Important:

- Under Windows, make sure that all backslash characters in all paths are doubled as in ****. Example:

```
conf.dir=C:\\Program\Files\\IBM\Application\Discovery\Batch\Server/conf
```

- Under Linux, remember that all paths use **/** as separators and spaces are escaped by using ****. Example:

```
conf.dir=/home/username/IBM\Application\Discovery\Batch\Server/conf
```

b) Enter the path to the directory where data files are stored; the default path is `${install.dir}/data`; if no other path is specified the default path is used.

```
## path to the directory where data files are placed
## default: ${install.dir}/data
data.dir={install.dir}/data
```

Important:

- Under Windows, make sure that all backslash characters in all paths are doubled as in ****. Example:

```
data.dir=C:\\Program\Files\\IBM\Application\Discovery\Batch\Server/data
```

- Under Linux, remember that all paths use **/** as separators and spaces are escaped by using ****. Example:

```
data.dir=/home/username/IBM\Application\Discovery\Batch\Server/data
```

c) Enter the path to the directory where temporary data is stored; the default path is `${data.dir}/tmp`; if no other path is specified the default path is used. Example:

```
## path to the directory where temporary data is placed
## default: ${data.dir}/tmp
tmp.dir={data.dir}/tmp
```

Important:

- Under Windows, make sure that all backslash characters in all paths are doubled as in ****. Example:

```
tmp.dir=C:\\Program\Files\\IBM\Application\Discovery\Batch\Server/tmp
```

- Under Linux, remember that all paths use **/** as separators and spaces are escaped by using ****. Example:

```
tmp.dir=/home/username/IBM\Application\Discovery\Batch\Server/tmp
```

Project Properties

The `project.properties` file contains a set of global settings, followed by the specific setting(s) for each type of component.

For each available component, there can be one or more configuration files. In `project.properties` file we need to enter the paths to these configurations files for each component. Additionally, in these files you can configure project specific settings. For example, you can configure **IBM AD Batch Server** to run only **Rule Based** reports for a specific project. Or you can configure **IBM AD Batch Server** to save the reports for one project in one folder and for other project in another folder. For more optional settings, see the project specific settings in the corresponding section for each component.

Global Settings

The global settings specify the projects on which the **IBM AD Batch Server** will operate and which components will run on the specified projects.

The `project.properties` file is where you can set the following global properties.

- Comma separated list of project names that will be the only ones considered for this service.

  Important: If no value is set for this parameter no report will be generated. ***** means all projects.

  ```
  projects.whitelist=<Project>
  ```

- Comma separated list of project names that will be excluded for this service (lower priority than whitelist). Leave empty or remove property if not needed.

  ```
  projects.blacklist=<Project>
  ```

- Comma separated list of component names that will be the only ones considered for this service. Ex.: **index** must be added as a component.

  ```
  components=index,gdbImport,annUpdate
  ```

  Optional components can be considered for this service.

Table 5. Optional Components

Component	Description
ruleBased	The **Rule Based** component generates reports for the resources specified in the configuration files according to the rules and parameters defined in the corresponding configuration files. **Note:** If the **Rule Based** component is used, make sure that the `ruleBased.properties` file is configured. For more information, go to “ruleBased.properties File” on page 87.
reports	The **Reports** component is used to generate the complexity reports. For more information, go to “Complexity Reports” on page 359.
cobolPP	The **cobolPP**, **jclPP**, and **pl1PP** components generate the expanded sources for Cobol, JCL, and PL/I. For more information, go to “View Expanded Source” on page 318.
jclPP	
pl1PP	
wsmetrics	The **wsmetrics** component is needed only if IBM ADDI Extension is used on the system. **Note:** Additionally, in order for the **wsmetrics** component to be executed, make sure that the **gdbImport** component is included in the components list of this service.

Important: It is vital that you set the parameters for the **Search Index**, **Annotations** and **GraphDB** components. Failing to do so will make some of the analysis impossible to generate in **IBM AD Analyze**.

Annotations

IBM AD Batch Server reads the configurations present under Annotations Database configurations from **IBM AD Configuration Server**. The configurations were set in `server.properties` in previous versions, and according to those settings it creates, if necessary, the repository where all user-defined annotations for all projects are stored.

Note: To make sure that Annotations Database was created in your database repository server instance check the following steps.

1. Go to **IBM AD Batch Server** installation folder and search for the `conf` folder.
2. Open the `server.properties` file that can be found in `conf` folder and verify that **IBM AD Configuration Server** host and environment are correctly filled in. For more information, see "Server Properties" on page 78.
3. Go to **IBM AD Configuration Server** > **Environment** > **Configurations** > **Annotations Database** and make sure that **Database name**, **Database schema**, and **Database server** settings are correctly filled in.
4. Start IBM AD Batch Server.
5. Go to the database repository instance set-up under Annotations Database configurations from **IBM AD Configuration Server** and you should find the database having the name mentioned in Database name field.

   Note: By default, **IBM AD Batch Server** will preserve in Annotations Database only those annotations for projects which are present in the projects list and will delete all other references/annotations. If you want to keep all annotations for all projects including the projects which are deleted at some point, uncomment and enter `false`.

```
## if true, all references of projects that do not appear in projects list will be deleted from annotations database.
(the default behavior)
```

```
#annUpdate.manageDeletedProjects=true
```

Application Settings

This setting allows the user to specify which components will run on a specific project.

Application Project Settings

This is the list of components that will execute for the project. **Empty** means no component will be executed; use ***** for all. Use a separate line for each project for which you want to define the components list.

```
#project.<projectName>.components=
```

Index Project Files

The **Index** component indexes the resources of a project so that a search in resources can be performed in **IBM AD Analyze** client, by using **Search in Files** analysis.

Important: Make sure that the Index location is a shared folder and has the appropriate permissions, for the situation when **IBM AD File Service** is installed and configured on a different machine than the Index location so that the **Search in Files** analysis can be used by all **Analyze Client** users. For more information, see the "Search in Files Analysis Tab" on page 316 section.

Index Global Settings

The Index component indexes the resources of a project so that a Search in resources can be performed in **IBM AD Analyze Client**, by using Search in Files analysis.

```
index.indexFolder=\\\\path\\<Folder>\\Index
```

Note:

- Starting with IBM AD V5.1.0.3, **IBM AD Search Service** is responsible with the access to the indexed data. Whether the authorization/authentication feature is used or not, the folder path in which the indexes are generated needs to be accessible both for **IBM AD Batch Server** and **IBM AD Search Service**. The path where the index data is generated needs to be added in the `conf.yaml` file of the **IBM AD Search Service**, where the *indexPath* parameter is present.
- For the accessible path defined in the `project.properties` file, backslashes must be doubled (****) and spaces in the path must have a single backslash as a prefix.

  Examples for the *index.indexFolder* parameter:

- `index.indexFolder=\\\\path\\<Folder>\\Index`
- `index.indexFolder=C:\\ibm\\Index`

Important:

- The folder path in which the indexes are generated needs to be accessible both for **IBM AD Batch Server** and **IBM AD Search Service**.
- The Index location is used during the configuration process of **IBM AD Search Service**.

The **`index.encoding`** parameter can be used to specify the encoding that is used to read target files. If this parameter is not configured in the `project.properties` file, the default encoding is UTF-8. You can set the value to another encoding according to IANA standards. This setting applies only to IBM AD Batch Server.

```
index.encoding=UTF-8
```



Warning: The encoding must match the one that is used in source files. For more information about how to configure the source file encoding, see the Chapter 5, “Configuring,” on page 33 section.

Once the indexes are created in the default location, if there is a need to change this default folder, the following parameter should be used with value **true**.

Important: Starting with IBM AD V5.1.0.3, the `index.forceFolder` parameter is no longer present in the `project.properties` configuration file. The **IBM AD Search Service** is now responsible with the access of the indexed data when Search in files analysis from **IBM AD Analyze Client** is performed. If you want to change the index folder path you can modify the *index.indexFolder* parameter, but in the same time you need to make sure that the same path is used in the `conf.yaml` file of the **IBM AD Search Service**, where the *indexPath* parameter is present.

```
index.forceFolder=true
```

If a folder matches one of the patterns that are specified below, it will not be indexed.

```
index.exclude.folderPattern=pattern,(pattern)...
```

If a file matches one of the patterns that are specified below, it will not be indexed.

```
index.exclude.filePattern=pattern,(pattern)...
```

Index Project Settings

All the parameters in the global settings can also be configured individually for each project. The parameters set for each project override the global parameters.

The output folder where the index for this project is stored must be unique per project.

Note: This setting does not override database entry!

```
project.<projectName>.index.indexFolder=
```

This is how you can override the database folder and create the index in a new location:

Note: Starting with IBM® AD V5.1.0.3, the `project.<projectName>.index.forceFolder` parameter is no longer present in the `project.properties` configuration file. The **IBM AD Search Service** is now responsible with the access of the indexed data when Search in files analysis from**IBM AD Analyze Client** is performed. If you want to change the index folder path for each project, you can modify the *project.<projectName>.index.indexFolder* parameter, but in the same time you need to make sure that the same path is used in the `conf.yaml` file of the **IBM AD Search Service**, where the *projects* section is present.

```
project.<projectName>.index.forceFolder=(true|false) default false
```

This is how you can specify extra folders to be indexed (apart from project folder):

Note: Make sure that all backslash characters in all paths are doubled: ****.

```
project.<projectName>.index.extraSources=path, path
```

If a folder matches one of these patterns, it will not be indexed.

```
project.<projectName>.index.exclude.folderPattern=pattern(,pattern)...
```

If a file matches one of these patterns, it will not be indexed.

```
project.<projectName>.index.exclude.filePattern=pattern(,pattern)...
```

ADI Indexing

The **ADI Index** component will index the resources of a project so that the **Business Rules Discovery (BRD)** feature, from **IBM® ADDI Extension** application, can display code snippets.

1. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\conf folder, open the `adi.properties` file, and make sure that the file contains the following URL for the ADI indexing service.

   ```
   ##URL for the ADI indexing service
   es.serverUrl=https://localhost:9444/addi/es
   ```

   Note: Please do not update the ADI indexing service URL, unless otherwise instructed by the Admin that sets-up and configures the IBM ADDI extension.
2. In the `project.properties` file, which is located in the `\conf` folder, enter the **adidx** component name.

   ```
   ## list of components that will be executed
   ## example components:
   ruleBased,reports,index,gdbImport,annUpdate,cobolPP,jclPP,pl1PP,wsmetrics,adidx
   ## empty means none will execute!!, use * for all
   components=adidx
   ```

Cobol, JCL and PL/I Preprocessing

View expanded source functionality in **IBM AD Analyze** can only be available if a preprocessing operation has been performed on the analyzed resources. Preprocessing of **Cobol** resources implies expanding all include files and applying all **REPLACE** compiler directives. For **JCL**, pre-processing implies expanding all **JCL** procedures and **JCL** include files.

Preprocessing of **PL/I** resources implies expanding all *include* files, having the following supported syntax:

```
File inclusion: %INCLUDE File, %XINCLUDE File, ++INCLUDE File, -INC File;
```

```
Library member inclusion: %INCLUDE MEMBER, %INCLUDE LIB (MEMBER).
```

The **Cobol**, **JCL** and **PL/I** preprocessing parameters allow the user to customize the details of the operation.

Cobol Preprocessor Global Settings

This parameter specifies the encoding used to read **Cobol** programs and copybooks when generating the preprocessed source. The default value is **UTF-8**, but any Java encoding is accepted. For example, for Japanese ideograms use **Shift_JIS**.

```
cobolPP.encoding=Shift_JIS
```

If the source code contains tabs instead of blank spaces, the preprocessor should know how many spaces to put instead of the tabs. The tab size default value is *4*, allowed values are *>= 1*.

```
#cobolPP.tabSize=4
```

Threads parameter is used to set the maximum number of threads per task to execute when preprocessing. The default value is *5*, allowed values are *>= 1*.

```
#cobolPP.threads=5
```

Emulate prefix parameter dictates whether or not to handle **PREFIX** and **-INC** directives when preprocessing, the default value is **true**.

```
#cobolPP.emulatePREFIX=true
```

Full path parameter dictates whether to specify in the inserted comment the full path of the expanded copy file in the preprocessed source, default value **true**. If set to **false**, only the name of the copy file and its content will be inserted in the preprocessed source.

```
#cobolPP.copy.fullPath=true
```

JCL Preprocessor Global Settings

This parameter specifies the encoding used to read files in order to generate preprocessed source. The default value is **UTF-8**, but any Java encoding is accepted. For example, for Japanese ideograms, use **Shift_JIS**.

```
jclPP.encoding=Shift_JIS
```

Threads parameter can be used to set the maximum number of threads per task to execute when preprocessing. The default value is *5*, but the allowed values are *>= 1*.

```
#jclPP.threads=5
```

PL/I Preprocessor Global Settings

This parameter specifies the encoding used to read files in order to generate preprocessed source. The default value is *UTF-8*, but any Java encoding is accepted. For example, in case of Japanese ideograms, use *Shift_JIS*.

```
pl1PP.encoding=Shift_JIS
```

Threads parameter can be used to set the maximum number of threads per task to execute when preprocessing. The default value is *5*, but the allowed values are *>= 1*.

```
pl1PP.threads=5
```

Important: If the source code contains characters or digits beyond the column 72 (in the inactive zone, for example *BG100 : PROCEDURE OPTIONS (MAIN);[ ]* ***00020002***) then the next additional setting must be made:

```
pl1PP.line_max_len=73
```

Rule Based Analysis Reports

The **Rule Based** component will generate reports for the resources specified in the configuration files according to the rules and parameters defined in the corresponding configuration files.

A configuration file is line based text organized as key/value pairs. General properties of a configuration file include:

- empty lines are ignored.

- lines that start with **#** are ignored (comments).
- keys do not contain spaces or equal sign **=**.
- a value starts at the first non-space character after **=**and finishes at the end of the line.
- values can expand on multiple lines by means of a single backslash character followed by newline.

Rule Based Global Settings

File specifying what information will be displayed on the first page of the generated reports.

```
ruleBased.configFile=ruleBasedConfig.properties
```

File containing the actual definition of the rules.

```
ruleBased.rulesFile=ruleBased.properties
```

Encoding of the file containing the actual definition of the rules, the default encoding is **UTF-8**, to modify, uncomment and enter desired encoding.

```
ruleBased.rulesFile.encoding=UTF-8
```

Option to generate an individual report for each source file where a rule failed, default value **true**.

```
ruleBased.reportsPerSource=true
```

Option to generate for each source file an additional file (named `controlTotal`) containing the number of rules that failed for that file. Valid only if the previous parameter, **`reportsPerSource`**, is set to **true**. The default value is **true**.

```
ruleBased.generateTotals=true
```

Output folder for rule based reports generated by source based rules. The default folder is the data folder, to set a different folder, uncomment and enter the path to that folder. The default format for the reports is `*.`PDF.

```
ruleBased.reportsFolder=
```

Output folder for reports generated in `*.`CSV format. The default folder is the data folder, to set a different one, uncomment and enter the path to that folder.

```
ruleBased.csvFolder=
```

Output folder for `controlTotals` files generated by source based rules. The `controlTotals` files are the files containing the number of rules that failed for a specific source file. The default folder is the data folder, to set a different folder, uncomment and enter the path to that folder. This parameter is only valid only if `generateTotals` is set to **true**.

```
ruleBased.totalsFolder=
```

Default suffix for the `controlTotals` file. To set a different suffix, uncomment and enter the desired suffix.

```
ruleBased.totalsSuffix=.CDGSTDS.TOTALS
```

Prefix for the `controlTotals` file, by default the `controlTotals` do not carry a prefix, to use a prefix uncomment and enter the desired prefix.

```
ruleBased.totalsPrefix=
```

Rule Based Project Settings

All the parameters in the global settings can also be configured individually for each project. The parameters set for each project override the global parameters.

- Absolute path to rule configuration file used for Rule Based analysis component.

```
project.<projectName>.ruleBased.rulesFile=projectName.rules.properties
```

- Option to generate an individual report for each source file where a rule failed. The default value is **true**.

```
project.<projectName>.ruleBased.reportsPerSource=
```

- Option to generate for each source file an additional file (named controlTotal) containing the number of rules that failed for that file. Valid only if **`reportsPerSource`** is set to **true**.

```
project.<projectName>.ruleBased.generateTotals=
```

- Output folder for rule based reports generated by source based rules. The default folder is the data folder. To set a different folder, uncomment and enter the path to that folder. The default format is `*.`PDF:

```
project.<projectName>.ruleBased.reportsFolder=
```

- Output folder for `*.`CSV files generated by data based rules. The default folder is the data folder. To set a different folder, uncomment and enter the path to that folder:

```
project.<projectName>.ruleBased.csvFolder=
```

- Output folder for controlTotals files generated by source based rules. The default folder is the data folder. To set a different folder, uncomment and enter the path to that folder. Valid only if **`generateTotals`** is set to **true**.

```
project.<projectName>.ruleBased.totalsFolder=
```

ruleBasedConfig.properties File

All data set in this file will appear on the cover page of the reports.

- Name of the owner of the report.

```
owner=
```

- Name of the organization.

```
organization=
```

- Role of report owner within the organization.

```
organizationRole=
```

- Image that will appear on cover page of the report.

```
coverImage=
```

- Project name that will appear on cover page; if not set here, the name of the project given as input will be used.

```
projectName=
```

ruleBased.properties File

This file specifies the actual rules on which the **Rule Based** reports are generated, different parameters for the rules as well as the source files on which the **Rule Based** analysis will run.

Here are some specific syntax elements of the `RuleBased.properties` file:

- <number>: is a positive integer.
- <identifier>: alphanumeric text with no spaces.
- <path>: absolute path to a file on disk.
- <class>: qualified class name for a rule.
- <text>: any string of characters on one line. Line continuation (single backslash at the end of line) can be used for multiple lines. A syntax comment is introduced hereby. It is not part of the file format.
- {key}: the value of the key as found in the file.

Rules can be parametrized, inputs being one of the optional parameters which can be set. Input values can be a list of data from the database based on a query. Rule characteristics:

- **id** - unique number to identify an input; it is used to create all subsequent keys for current input.
- **key** - unique key used for substitution in query rules.
- **name** – input name.
- **query** - query to obtain a list of values from the repository which will act as parameters.
- **firstColumnType** - declare the type for the first column of the query. The default value is numeric. Accepted, string.

Parameter values can be simple values (numbers or text) and are also optional. Parameter characteristics:

- **id** - unique number to identify a parameter; it is used to create all subsequent keys for current parameter.
- **key** - unique key used for substitution in query rules.
- **name** – parameter name.
- **pattern** - if parameter is a simple value, pattern can be used to validate the data the user entered.
- **length** - if parameter is a simple text value, length can be used to verify the max length of the data.
- **value.lessThan** - for numeric simple values it can be used for validation.
- **value.greaterThan** - for numeric simple values it can be used for validation.
- **prefix** - text to prefix the user chosen data (transforms the parameter as text).
- **suffix** - text to be appended to the user chosen data (transforms the parameter as text).

Rules can be organized in groups according to specific criteria. Rule group characteristics:

- **id** - uniquely identifies a group; it is used to create all subsequent keys for current group.
- **name** - name to display for the group.
- **parent** - a {group.id} value of another group that contains this group.
- **description** - additional information on the group displayed in the application.

Rules are the basis of the Rule Based analysis action. Rule characteristics include:

- **id** -unique number that identifies a rule; it is used to create all keys for current rule.
- **name** – the name that is visible for the user. This will appear in the report.
- **sourceBased** - classifies the rule. The report will be generated in `*.`PDF format if **`sourceBased`** is set to **true**. If **`sourceBased`** is set to **false**, the report will be generated in `*.`CSV format. The default value is **true**.
- **weight** – a number used to calculate the weight of a certain rule, appears in the report summary. The default value is **1**.
- **query** and **selectiveQuery** - paths to query files.
- **class** - qualified class name for a rule implementation.
- **groups** - list of group ids that will show the rule.
- **description** - long text that describes the rule to the end-user.

- **params** - list of parameter keys that are required by this rule.
- **inputs** - list of input keys that are required by this rule.

Reports

The **Report** component will generate the reports specified in the configuration file.

Report: Global Settings

Important: The global settings used by these reports is set in the `project.properties` file.

Absolute path to the file containing a description of the report headers: `#reports.configFile=${install.dir}.reportsConfig.properties`.

Comma separated list of report codes that will be generated.

```
## 15103=Halstead
## 15104=Cyclomatic
## 15105=Maintainability
## 15106=Heuristic
## 15108=Unused Routines Within Programs
#reports.whitelist=
```

Comma separated list of report codes that will NOT be generated (lower priority than whitelist).

```
#reports.blacklist=
```

Output folder for generated reports. The default folder is the data folder, to change, uncomment and enter absolute path to desired folder.

```
#reports.outputFolder=
```

Option to generate a unique report for all resources. The default value is **true**. To generate a separate report for each resource, enter **false**.

```
#reports.coalesceInputs=true
```

Report: Project Settings

Important: The global settings used by these reports is set in the `project.properties` file.

All the parameters in the global settings can also be configured individually for each project. The parameters set for each project override the global parameters.

Comma separated list of report codes that will be generated for the project.

```
## 15103=Halstead
## 15104=Cyclomatic
## 15105=Maintainability
## 15106=Heuristic
## 15108=Unused Routines Within Programs
#project.<projectName>.reports.whitelist=
```

Comma separated list of report codes that will NOT be generated (lower priority than whitelist)

```
#project.<projectName>.reports.blacklist=
```

Absolute path to custom report configuration file for the project.

```
#project.<projectName>.reports.configFile==${install.dir}.projectName.reportsconfig.properties
```

Option to generate a unique report for all resources in the specified project. The default value is **true**. To generate a separate report for each resource, enter **false**.

```
#project.<projectName>.report.coalesceInputs=<true|false>
```

reportsConfig.properties File

This file allows the user to determine what information will appear on the first page of the reports, whether to show the appendix of a report, as well as where to find the configuration files which allow the customization of different report types.

The data set in this file will appear on the cover page of the reports.

- Name of the owner of the report.

```
owner=
```

- Name of the organization.

```
organization=
```

- Role of report owner within the organization.

```
organizationRole=
```

- Image that will appear on cover page of the report.

```
coverImage=
```

- Project name that will appear on cover page; if not set here, the name of the project given as input it will be used.

```
projectName=
```

- Show appendix in reports; the default value is **false**.

```
showAppendix=
```

- Absolute path to the file containing custom settings for reports (values for weights, colors for pie charts, etc.)

```
halstead.settingsFile=
heuristic.settingsFile=
cyclomatic.settingsFile=
maintainability.settingsFile=
unusedRoutines.settingsFile=
```

cyclomatic.properties File

This file contains settings which allow the customization of the **Cyclomatic** reports.

Note: This configuration file is located in the `conf/reports` folder.

The colors used in the pie charts present in the report:

```
#simpleColorPreference=0,255,0
#mediumColorPreference=250,196,69
#complexColorPreference=221,0,0
```

Values for thresholds used to separate simple/medium/complex groups:

```
#simpleThresholdPreference=10
#mediumThresholdPreference=50
```

halstead.properties File

This file contains settings which allow the customization of the **Halstead** reports.

Note: This configuration file is located in the `conf/reports` folder.

By default, all statements are considered operators. If you want to set a specific statement as operand, enter the statement id (from the **Statement** table in the database) then enter **false**.

```
##stmt id = check state
#if not present, falls to default. In default file, default value = check
#29=false
#28=false
```

heuristic.properties File

This file contains settings which allow the customization of the **Heuristic** reports.

Note: This configuration file is located in the `conf/reports` folder.

Colors used in pie charts present in the report:

```
#simpleColorPreference=0,255,0
#mediumColorPreference=250,196,69
#complexColorPreference=221,0,0
```

Values for thresholds used to define simple/medium/complex groups:

```
#simpleThresholdPreference=200
#mediumThresholdPreference=500
```

By default, all statements are assigned a weight of **1**, if you want to assign a different weight for a specific statement, enter the statement id (from the **Statement** table in the database) and then desired weight value.

```
##stmt id = weight value
#if not present, falls to default. In default file, default value = 1
#1815=100
#1814=1000
#109=2
#108=3
```

maintainability.properties File

This file contains settings which allow the customization of the **Maintainability** reports.

Note: This configuration file is located in the `conf/reports` folder.

Colors used in pie charts present in the report:

```
#complexColorPreference=0,255,0
#simpleColorPreference=221,0,0
#mediumColorPreference=250,196,69
```

Values for thresholds used to define simple/medium/complex groups:

```
#simpleThresholdPreference=65
#mediumThresholdPreference=85
```

By default, all statements are considered operators. If you want to set a specific statement as operand, enter the statement id (from the **Statement** table in the database) then enter **false**.

```
#stmt id=false
#29=false
#28=false
```

unusedRoutines.properties File

This file contains settings which allow the customization of the **Unused Routines** reports.

Note: This configuration file is located in the `conf/reports` folder.

By default, **Exit Procedures** will NOT be shown in the report, to include the **Exit Procedures** in the generated report, uncomment and enter **false**.

```
#filterExitProcedures=true
```

Web Service Metrics Component

Web Service Metrics component generates input data for **IBM AD Web Service**. The metrics values are packaged into **JSON** objects.

IBM AD Web Service

IBM AD Web Service collects the data provided by **Web Service Metrics** component and prepares it for delivery.

IBM AD Web Service parameters are stored in a `webservice.properties` file available in **IBM AD Batch Server** installation folder.

The following parameters can be set:

1. The IP of the computer where **IBM AD Web Service** is installed. The default value is **0.0.0.0** (all interfaces). Another valid value is the external IP of the computer.

   ```
   ## web service host
   #host=
   ```

2. The port no. to be used by the **IBM AD Web Service**; the default port is **8090**, if no other value is specified the default value will be used.

   ```
   ## web server port
   ## default: 8090
   #port=
   ```

3. The administrative port number to be used by **IBM AD Web Service**; the default administrative port is **8099**, if no other value is specified the default value will be used.

   ```
   ## web service administrative port
   ## default: 8099
   #admin-port=
   ```

4. Absolute path to the **Web service** directory.

   ```
   ## web service directory
   web-dir=webdata
   ```

5. Absolute path to the **JSON** files directory.

   ```
   ## json files directory
   rest-dir=wsmetrics
   ```

6. Absolute path to file containing the Web service authentication data.

   ```
   ## authentication info for the web service
   auth-file=auth.properties
   ```

7. Absolute path to file containing the SSL certificate.

   ```
   ##ssl keystore file
   Keystore-file=keystore.jkl
   ```

8. Absolute path to the file containing log configuration data.

   ```
   ## log configuration file
   log-conf-file=webservice.log4j.properties
   ```

IDCAMS Preprocessor

To display the dataset usage via **IDCAMS** utility in **Dataset Usage** in **Jobs**, a preprocessing operation must be performed on the analyzed resources. Running **IDCAMS** preprocessing implies saving data from all available **JCL** jobs that execute **IDCAMS** utility into the database repository project repository. **IDCAMS** preprocessing does not need to be added as a separate component, it runs on each project added in the <project.whitelist>.

Global Settings

This parameter specifies the encoding used to read **JCL** jobs and **CTRL** files when parsing the sources for all available projects. The default value is **UTF-8**, but any Java encoding is accepted.

You can setup this encoding in the `project.properties` file.

For example, in case of Japanese ideograms use **Shift_JIS**:

```
ams.encoding=Shift_JIS
```

Project Settings

This parameter specifies the encoding used to read **JCL** jobs and **CTRL** files when parsing the sources for a specific project. The default value is **UTF-8**, but any Java encoding is accepted.

You can setup this encoding in the `project.properties` file, using the following entry:

```
project.<projectName>.ams.encoding=UTF-8
```

Importing Multiple Projects when IBM AD Batch Server is Installed and Configured on Linux/zLinux

Perform the following configurations if you want to import multiple projects when **IBM AD Batch Server** is installed and configured on Linux/zLinux.

Important: The following configurations need to be performed for versions up to **IBM AD V5.1.0.7** as follows:

By modifying the Global Settings

1. On the machine where IBM AD Batch Service is installed, edit `/etc/sysctl.conf` file and add the following line.

   ```
   fs.file-max = 65536
   ```

2. Apply the changes by using the following command.

   ```
   sudo sysctl -p /etc/sysctl.conf
   ```

Per user limit

1. Edit as root the following system configuration file.

   ```
   % sudo vi /etc/security/limits.conf
   ```

2. Modify the values for **AD_user**.

   Note: We assume that **JBOSS** was launched with the system user **AD_user**.

   ```
   AD_user           soft    nofile          4096
   AD_user           hard    nofile          8192
   ```

3. If you want to raise the limits for all users, modify the values as follows.

   ```
   *           soft    nofile          4096
   *           hard    nofile          8192
   ```

4. Open a new session with the user that starts **OrientDB** and check whether the changes are valid.

```
% su AD_user (orient_user/batch_user)
% ulimit -n
1024
```

5. If the changes were not taken into account perform the following actions:

   a. Edit `/etc/pam.d/su:` as follows.

   ```
   % sudo vi /etc/pam.d/su
   ```

   b. Uncomment the following line.

   ```
   session    required   pam_limits.so
   ```

   c. The changes are taken into account at the next login.

   ```
   % su AD_user
   % ulimit -n
   4096
   ```

STEP 9. Configuring IBM AD Cross Applications Service

IBM AD Cross Applications Service is an **additional** service that needs to be configured to show calls between different mainframe projects that have their databases on the same DB instance in **IBM AD Analyze Client**.

Follow the configuration steps that are needed to have up and running **IBM AD Cross Applications Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Cross Applications Service available in IBM AD Configuration Server
3. Restart IBM AD Cross Applications Service

1. Configure the parameters that are present in the `conf.yaml` file

Important: The configuration of the following parameters is not mandatory. It is recommended to follow below steps only in case that you had previously configured these parameters and you performed an upgrade to the latest version of **IBM AD** product.

On the machine where **IBM AD Cross Applications Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Add the port of **IBM AD Configuration Server**.

```
## Coordination and Configuration Server port
## default 2181
ccs.server.port: 2181
```

2. Set the **https** parameter as follows:

a. If the **https** parameter is set to *false*, a non-secured communication is used.

```
#if communication should be secured with TLS
https: false
```

b. If the **https** parameter is set to *true*, a secured communication is used.

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
#if communication should be secured with TLS
https: true
```

Note: If the **https** parameter is set to *true*, an additional step needs to be performed.
Locate `startServer.bat` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/` and replace the following line:

```
set tlsoptions=
```

with:

```
SET keystorepath=<"path_to_keystore">
SET keystorepass=<"password_of_keystore">
set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
-Djavax.net.ssl.keyStorePassword="%keystorepass%"
```

Where:

- **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Cross Applications Service**.
- **Keystore password** is the keystore password.

3. Leave blank the line where the **authSrv** parameter is present since **Authentication Server (DEX)** is not needed.

```
#authentication server URL
authSrv:
```

4. The default value of the **disableAuth** parameter is *true*. Leave the default value since **Authentication Server (DEX)** is not needed.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: true
```

5. Optionally, you can decide what project resolutions to display in the analysis, when more projects contain the same program definition.

```
#optional: force resolution target
#program name pattern from source project will be solved only in target project
restrictions:
 - pattern:  "Program1"
   from.project: Project1
   to.project: Project2
```

Where:

- **pattern** - represents the name of the program which has the source present in other projects.
- **from.project** - represents the project that contains the program without the source.
- **to.project** - represents the project resolution that you decide to show in the analysis.

2. Make IBM AD Cross Applications Service available in IBM AD Configuration Server

By adding a restriction in the **Cross Applications Service** page, you can decide what project resolutions to display in the analysis, when more projects contain the same program definition.

After **IBM AD Cross Applications Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Cross Applications Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Cross Applications Service**. The **Cross Applications Service** settings page is displayed.
2. To add a new restriction, click the **Add** button. As a result, the **Add Restriction** page is displayed.
3. In the **Pattern** field, enter the name of the program that has the source present in other projects.
4. In the **From Project** field, choose the project that contains the program without the source
5. In the **To Project** field, choose the project resolution that you decide to show in the analysis.
6. Click **Save**.

3. Restart IBM AD Cross Applications Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Cross Applications Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. If the service does not start, check the `cross.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/log` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/` and locate the `startServer.sh` file.
  2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x startServer.sh
     ```
  3. If the service does not start, check the `cross.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/log` folder.

Important: At this installation and configuration point, everything is put in place for having analysis available in **IBM AD Analyze Client**. To take full advantage of the available analysis functionality, you need to install **IBM AD Analyze Client**. For more information see, "Installing IBM AD Analyze Client" on page 27.

STEP 10. (Optional) Configuring IBM AD Analyze Server

This step is optional and not necessary unless Java source code will be included in one or more projects in this environment.

On Linux only, Open the `servermount.properties` file, which is located under `<installation folder>\IBM Application Discovery Batch Server/conf` folder and specify how the windows shared folders are mounted on the local files system, by using the following pattern:

```
\\\\machine IP\\WindowsSharedFolder=/home/user/LinuxFolder
```

Example:

```
\\\\192.168.56.57\\ProjectsSharedPathWindows=/home/user/ ProjectsSharedPathLinux
```

It is mandatory to mount the Remote Path from **Analyze Server Manager** > **Server Settings**.

Following are the configuration steps that are needed after **IBM AD Analyze Server** was installed.

- **Under Windows**: to access the configuration parameters, select **Start** > **Programs** > **IBM Application Discovery Analyze Server** > **IBM Application Discovery Analyze Server Manager**.
- **Under Linux**: to access the configuration parameters, go to `<Installation Path>\IBM Application Discovery Servers\IBM Application Discovery Analyze Server` and run `manager.sh`.

In the **Server settings** tab, the **Server properties** and **Server arguments** sections display default data that was entered when **IBM AD Analyze Server** was installed.

To configure **IBM AD Analyze Server**, follow the instructions below.

1. In **IBM AD Configuration Server**, go to **Configurations** > **Default** > **Analyze Servers** and fill in the following details:
   - **`Host`**: IP / Hostname where **IBM AD Analyze Server** is installed.
   - **`RMI Registry Port`**: 1099 (Default).
2. Configuring the server database – the Database settings tab:

   Select the **Database Settings** tab. In the Location area, fill in the following parameters:

   - **Server type** field is completed by default with *SQL server*.
   - **Server IP**: Enter the IP of the computer where SQL Server is installed.
   - **Server port**: Server port is the access port, by default, 1433 port is used.

   Important: Make sure that the IP address and the port number you set here are the same as the ones entered in the **Relational database servers** page of **IBM AD Configuration Server**.

   Database instance, this field must be used in case the default database name was not chosen at SQL Server installation time. In the Authorization area, fill in the following parameters:

   - **Database name**: Enter a name for the database.
   - **User and password**: Give a user and password that can be used to create the database.

   After you completed the details of the database, click **Create database** to create the database with the selected parameters. If the database was configured correctly, after **Test database** is clicked, a message with the installed DB version will be displayed. Click **Save** to apply the settings.

   You can also select an existing database. If the selected database belongs to an older version of **IBM AD** and the database structure is now obsolete, a message is displayed indicating the current version of the selected database. The user is given the option to upgrade the existing database. Press **Upgrade** if applicable. After the upgrade process was performed, press again **Test database** to make sure that the upgraded database is functional. The version of the upgraded database is presented and **Upgrade** button is no longer available. For incomplete or corrupted databases one of the following messages may be displayed: **Database x is not a valid database** or **Cannot extract relevant data from the database. Database may be nonexistent, obsolete or invalid**.

   After modifying the settings in any of the tabs, do not forget to press **Save** to apply them. An asterisk at the beginning of the title of a tab indicates that parameters in that tab were modified but not saved.
3. Specifying allowed **IBM AD Analyze Clients**:

   Note: This configuration applies only to Java projects.

   - **IBM AD Analyze** clients can be of two types: manager and user.
   - **IBM AD Analyze** client of the **manager** type, can create shared projects, build shared projects, and delete shared projects.
   - **IBM AD Analyze** client of the “user” type, can only import the shared projects and perform analysis.

Manager and user types are server-related attributes, which means that a server determines the type for a client connecting to that server by looking up the client IP in the configuration files. This means that a client can be a manager on one server and a user on another server.

a) To add a Manager to the Managers list: Click **Add** in the upper right corner of the **Access Settings** tab: **New Access Data** dialog window is displayed. Enter the IP of the computer of the user who will access the server as a Manager (the type of owner is selected by default) then press **OK** to add the new manager to the list of Managers. To delete one of the Managers from the list, select it then press **Delete**. If you want to allow access to all the projects on the server to all potential users, do not add any users to the List of Users. If you want to limit the access to the projects to a number of specific users, select **restrict user IP** then add all of them to the List of Users. Only users in the List of Users and List of Managers will have access to the projects shared on the server.

b) To add a user to the List of Users, click **Add** from the List of Users area of the **Access Settings** tab: the New Access Data dialog window is displayed. Enter the IP of the computer of the user who will access the projects as a User (the type of owner is selected by default) then press OK to add the new user to the list of Users.

Under Windows: start **IBM AD Analyze Server**: Click **Start** and then select **All Programs** > **IBM Application Discovery Analyze Server** > **Start IBM Application Discovery Analyze Server**.

Alternatively, to start the server: From the Start menu, choose **Programs** > **IBM Application Discovery Analyze Server** > **Start IBM Application Discovery Analyze Server Monitor** then go to monitor icon from the taskbar, right-click on the icon, and select **Start service**. When the server is running, the green arrow from **Server Monitor** icon indicates that the server is started.

Under Linux: Go to `<Installation Path>\IBM Application Discovery Servers\IBM Application Discovery Analyze Server` and run `StartServer.sh`. Please make sure that this process remains alive.

Important: For monitoring the **IBM AD Analyze Server** tasks, see "Finding Log Files" on page 537.

With Authentication

The authentication process is conducted through **DEX** or through any other **Authentication Server** that supports the **OAuth2** protocol. **Authentication Server (DEX)**, which is delivered through the **IBM ADDI** installer, is an identity service that uses **OpenID Connect** and it is used in the following configurations.

When you start **IBM AD Analyze Client**, you need to login by using login button that is present on the right bottom of the screen to observe the list of available projects. A browser page is opened into which the user and password are filled in. The credentials reach the **Authentication Server (DEX)**, using the LDAP protocol, and it checks whether the credentials of the user are bounded to an account in **Secure Storage**. The **Secure Storage** can be an **Active Directory** or any other entity that stores users and groups and can communicate through LDAP. For more information, see the "Authorization and Authentication" on page 430 section.

Once authenticated, an authorization process is started to determine the list of available mainframe projects and whether the user can access all the source files that are used in the listed mainframe projects. Based on the configurations that are made, **File Service** maps the groups of users to a certain folder with the source files that can be on the same machine with **File Service** or not. Once you are authenticated and authorized, you can start the analysis on the available mainframe projects and on the source files.

STEP 1. Configuring IBM AD Configuration Server

The IBM AD Configuration Server component can run with the default settings. If the default settings are not compatible with your environment, you can configure the component and overwrite the default settings.

1. Configure the settings in the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/server.properties` file.
   a) Configure the port that AD Configuration Server listens on by setting the value of the **`server.port`** parameter.

The default value is 2181.

b) Configure the number of the snapshots and the corresponding logs that are retained by AD Configuration Server. To configure the setting, set the value of the **zookeeper.autopurge.snapRetainCount** parameter.

The default value is 4, and the minimum value is 3.

c) Configure the time interval in hours for the purge task by setting the value of the **zookeeper.autopurge.purgeInterval** parameter.

The purge task deletes old snapshots and the corresponding log files according to the time interval. The default value of the parameter is 24. To enable the purge task, you must set a value that is greater than 0.

2. Configure the web server settings in the <IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/webservice.log4j.properties file.

a) Configure the root logger level and the appenders by setting the **log4j.rootLogger** parameter with one of the following values:

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- ALL

The default log level is INFO. The default appenders are file, which indicates a rolling file appender, and stdout, which indicates a console appender.

Example

```
log4j.rootLogger=DEBUG
```

b) Configure the file roller appender log level by setting the **log4j.appender.file.threshold** parameter with one of the following values:

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- ALL

Note: If you do not set the value of the **log4j.appender.file.threshold** parameter, the file roller appender log level is the same as the root logger level. To set the **log4j.appender.file.threshold** parameter, the value must be lower than or equal to the root logger level.

Example

```
log4j.appender.file.threshold=ERROR
```

c) Configure the file roller appender location by setting the value of the **log4j.appender.file.File** parameter.

Note: The value of the **log4j.appender.file.File** parameter must be a valid absolute or relative path.

Example

```
log4j.appender.file.File=/home/user/logs/webservice.log
```

d) Configure the file roller appender minimum number of backup files to keep by setting the value of the **log4j.appender.file.MaxBackupIndex** parameter.

Example

```
 log4j.appender.file.MaxBackupIndex=5
```

e) Configure the file roller appender maximum file size by setting the value of the **log4j.appender.file.MaxFileSize** parameter.

Example

```
log4j.appender.file.MaxFileSize=100M
```

3. Configure the web server settings in the <IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/admin-ws.properties file.

   The web server is attached to AD Configuration Server.

   a) Configure the port that the web server listens on by setting the value of the **port** parameter.

   The default value is 8080.

   b) Configure the path to the configuration file of the logger by setting the value of the **log-conf-file** parameter.

   The default value is <IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/webservice.log4j.properties.

4. By default, the HTTP protocol is used to run the web service. To use the SSL/HTTPS protocol, follow the steps:

   a) Generate a self-signed key pair and store it in a Java keystore by using the Java Keytool command-line interface. Run the following command:

   Note: Java SDK or Java JRE must be installed, and the *JAVA_HOME* and *PATH* environment variables must be configured for the Java SDK or Java JRE.

```
keytool -genkeypair -keyalg RSA -alias {alias}
-ext SAN=DNS:localhost,IP:127.0.0.1 -dname {dname}
-validity {validity} -keysize 2048 -keypass {keypass}
-storepass {storepass} -keystore {keystore}
```

{alias}
The name that is used by the Java keystore to identify the generated key. The name must be unique within the Java keystore.

{dname}
The distinguished name from the X.500 standard. This name is associated with the alias for the key pair in the keystore. Also, the name is used as the value in the "issuer" and "subject" fields in the self-signed certificate.

{validity}
The number of days that the certificate that is attached to the key pair is valid.

{keypass}
The password that is needed to access the key pair within the keystore.

{storepass}
The password for the Java keystore.

{keystore}
The path to the keystore file, which is used to store the generated key pair. If the file does not exist, a keystore file is created.

Example

```
keytool -genkeypair -keyalg RSA -alias my-key-pair
-ext SAN=DNS:localhost,IP:127.0.0.1 -dname CN="IBM AD"
-validity 9999 -keysize 2048 -keypass my-key-password
-storepass my-store-password -keystore C:\my_keystore
```

b) Configure the web server that is attached to IBM AD Configuration Server to use the SSL/HTTPS protocol. In the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/keystore-config.properties` file, configure the following parameters:

`path`
Set the value to the path of the Java keystore that is generated in the preceding substep.

`storepass`
Set the value to the password for the Java keystore.

`keypass`
Set the value to the password that is needed to access the key pair within the keystore.

c) In the `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/conf/admin-ws.properties` file, set the value of the **`keystore-conf-file`** parameter to the path of the keystore configuration file.

5. Make sure that **IBM Application Discovery Configuration Service Admin** is up and running. If any error occurs, start the service manually:

- **On Windows**

a. Go to **Start Menu** > **IBM Application Discovery and Delivery Intelligence** and **Launch IBM Application Discovery Configuration Service Admin**.

b. If the service does not start, check the `server.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/` folder.

- **On Linux**

a. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/` and locate the `startServer.sh` and `startWebServerUI.sh` files.

b. Open a terminal under the current directory and run the following commands to launch the **IBM Application Discovery Configuration Service Admin**.

```
startServer.sh
startWebServerUI.sh
```

c. In case that the `.sh` files are not executable, open a terminal and run the following commands for flagging them as executable:

```
chmod +x startServer.sh
```

and

```
chmod +x startWebServerUI.sh
```

d. If the service does not start, check the `server.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Configuration Service/` folder.

Important: For monitoring the **IBM AD Configuration Server** tasks, see “Finding Log Files” on page 537.

STEP 2. Configuring IBM AD Web Services

IBM AD Catalog

The **IBM AD Catalog** package is used to enable the impact analysis on the APIs published by a **z/OS Connect** server.

Configuring the Catalog Service

Note: The Catalog Service uses only one database at a time and it can be created on **SQL Server** or **Db2 for LUW**.

Configuring the Catalog Service to run on SQL Server

By performing the following actions, the data is imported in the SQL database.

1. Open the `server.xml` file from `\wlp\usr\servers\ad_server`.
2. Enable **IBM AD Catalog**, by removing the `<!-- -->` comment around its `<webApplication>` element.

```
<webApplication contextRoot="com.ibm.etools.ad.catalog" id="com.ibm.etools.ad.catalog"
location="com.ibm.etools.ad.catalog.war" name="ad-catalog">
        <application-bnd>
            <security-role name="ADCatalogUser">
                <user name="catalog"/>
            </security-role>
        </application-bnd>
</webApplication>
```

3. Open the `server.xml` file from `\wlp\usr\servers\ad_server`.
4. In the *<!-- Add the Catalog database connection details-->* area, enter the SQL database in the **databaseName** field.

```
databaseName="Catalog"
```

5. In the same area, enter the IP of the server in the **serverName** field.

```
serverName=
```

6. Enter the SQL port in the **portNumber** field (the default is 1433).

```
portNumber="1433"
```

7. Enter the user and password used to connect to the SQL database in the **user** and **password** fields.

```
<!-- Add the Catalog database connection details-->
<dataSource id="ADCatRDB" jndiName="jdbc/ad/catalog/relational" type="javax.sql.DataSource">
    <jdbcDriver libraryRef="JTDSLib" javax.sql.DataSource="net.sourceforge.jtds.jdbcx.JtdsDataSource"/>
    <properties databaseName="catalog" serverName="127.0.0.1" portNumber="1433" user="sa" password="password"/>
</dataSource>
```

   Important: The password that is used to connect to the SQL database can be encrypted. The **securityUtility** command supports plain text encryption for Liberty. For more information, see securityUtility command.
8. Make sure that **IBM® AD Catalog Service** is configured in **IBM® AD Configuration Server**. For more information, see “ Configuring IBM AD Catalog Service” on page 47.
9. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

   Note:

   - Wait until the services are restarted, this can take a few minutes to complete.
   - The Catalog database will be automatically populated when **Data Collector** starts for the first time.

Configuring the Catalog Service to run on Db2 for LUW

By performing the following actions, the data is imported in the Db2 for LUW (Linux, UNIX, and Windows) database.

1. Open the `server.xml` file from `\wlp\usr\servers\ad_server`.

2. Enable **IBM AD Catalog**, by removing the <!-- --> comment around its <webApplication> element.

```
<webApplication contextRoot="com.ibm.etools.ad.catalog" id="com.ibm.etools.ad.catalog"
location="com.ibm.etools.ad.catalog.war" name="ad-catalog">
        <application-bnd>
            <security-role name="ADCatalogUser">
                <user name="catalog"/>
            </security-role>
        </application-bnd>
</webApplication>
```

3. In the *<!-- Add the Catalog database connection details-->* area, add the following library.

```
<library id="DB2JCCLib">
        <fileset dir="${shared.config.dir}" includes="db2jcc4-4.21.29.jar
com.ibm.ez.db2jcc.license.cisuz-1.0.0.jar"/>
</library>
```

Important: Make sure that in the `wlp/usr/shared/config` folder, the two `.jar` files are present. Depending on the Db2 version that is installed on your machine, the `.jar` file names can differ.

4. In the *<!-- Add the Catalog database connection details-->* area, edit the <dataSource> XML element as follows.

Example:

```
<dataSource id="db2" jndiName="jdbc/ad/catalog/relational">
        <jdbcDriver libraryRef="DB2JCCLib"/>
        <properties.db2.jcc databaseName="CATDB" serverName="localhost"
portNumber="50000" user="db2admin" password="{aes}ABitSh1jv1T9i8tpGN7uQiSvx+yAZl+
+igoVWxzzMu3CoI4lM6d69HP69MuRDeIEDQ=="/>
</dataSource>
```

Note: When configuring Catalog Service to run on Db2 for LUW, make sure that `properties.db2.jcc` is present in the <dataSource> XML element.

Where:

- **databaseName** - expects the Db2 database name.
- **serverName** - expects the name/IP of the server when Db2 Server is installed.
- **portNumber** - expects the port number configured for the communication with Db2 Server. The default value is 50000.
- **user** - expects the user that is used to connect to the Db2 database.
- **password** - expects the password that is used to connect to the Db2 database.

  Important: The password that is used to connect to the Db2 database can be encrypted. The **securityUtility** command supports plain text encryption for Liberty. For more information, see securityUtility command.

5. Make sure that **IBM AD Catalog Service** is configured in **IBM AD Configuration Server**. For more information, see “ Configuring IBM AD Catalog Service” on page 47.
6. Once the configuration is done, go to the **Dashboard** tab, in **IBM AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

Note:

- Wait until the services are restarted, this can take a few minutes to complete.
- The Catalog database will be automatically populated when **Data Collector** starts for the first time.

Configuring the Data Collector

Follow below steps to configure **Data Collector** that gathers the information related to the services exposed by the mainframe.

1. Copy the files for **Data Collector** from `zoscDataCollector` to a new working directory.

2. Update the `DC.properties` file in this newly created working directory, by setting the following property values:

   Important: Please do not update other property values in the file, unless otherwise instructed by IBM Support.

   a. **ZoscURL**: Specify the protocol, the host name and the port for the **z/OS Connect** server as a URL.

      Note: You need to append `/zosConnect/apis` as the path part of the URL. An example value is *http://<zoscHost>:9081/zosConnect/apis*.

   b. **ZoscUser**: Specify the user name by which **Data Collector** connects to the **z/OS Connect** server. If user authentication is disabled on **z/OS Connect** server or **IBM AD Catalog**, you do not need to define this property value in the property file

   c. **ZoscPass**: Specify the password (in plain text) used by **Data Collector** to connect to the **z/OS Connect** server. If user authentication is disabled on **z/OS Connect** server or **IBM AD Catalog**, you do not need to define this property value in the property file

   d. **DataCollectorId**: Specify the **Data Collector** ID, for **AD** to identify an instance of **z/OS Connect** server.

      Note: For v5.0.4, the **DataCollectorId** string must be the same as the **ZoscURL** property value. Additionally, **DataCollectorId** must contain the *zosConnect/apis* string which can be either *zosConnect/apis* or the full URL of the *ZoscURL*.

   e. **ADCatalogURL**: Specify the protocol, the host name and the port used by the **AD Catalog**.

      Note: You need to append `com.ibm.etools.ad.catalog/cat/entity` to the URL path. An example value is *http://<catalogHost>:9080/com.ibm.etools.ad.catalog/cat/entity/*.

   f. **ADCatalogUser**: Specify the user name by which **Data Collector** connects to the **AD Catalog** If user authentication is disabled on **z/OS Connect** server or **IBM AD Catalog**, you do not need to define this property value in the property file.

   g. **ADCatalogPass**: Specify the password (in plain text) used by **Data Collector** to connect to the **IBM AD Catalog** If user authentication is disabled on **z/OS Connect** server or **AD Catalog**, you do not need to define this property value in the property file.

3. Check the PATH environment variable. You need to add the directory where `java.exe` resides in the *PATH environment* variable, in case it is missing.
4. Update `zoscdc.cmd`.

   Important: You need to specify the jar file of **JSON4J** that resides under `/WAS_Liberty_Library/lib/` with full path name, as a part of the *-classpath* argument. The file name of the jar file should look like `com.ibm.json4j_`$x.x.x$`.jar`.

   Alternatively, you can specify the property values listed in **step 2** as the property values passed to the java command. For example, the **ZoscUser** property can be specified as an argument for **-DZoscUser=user1**.

How to Collect z/OS Connect API Information

Take the following steps to invoke **Data Collector** to collect **z/OS Connect** API information.

1. Open a windows command prompt and change the current directory to the `/Data_Collector_Working_Directory/`.
2. Run `zoscdc.cmd`.

IBM AD Audit

Configuring IBM AD Components to Use the Audit Service

IBM AD Build Client, **IBM AD Batch Server** and **IBM AD Analyze Client** can be configured to use the **Audit Service**.

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **General** tab.
2. The **Audit Service** for **IBM AD Batch Server** is set to be enabled by default. To stop using the **Audit Service**, set **`audit.enable=true`** to **`audit.enable=false`** in the `server.properties` file located in the `\conf` folder.

```
##Audit server
# default true
#audit.enable=true
```

3. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

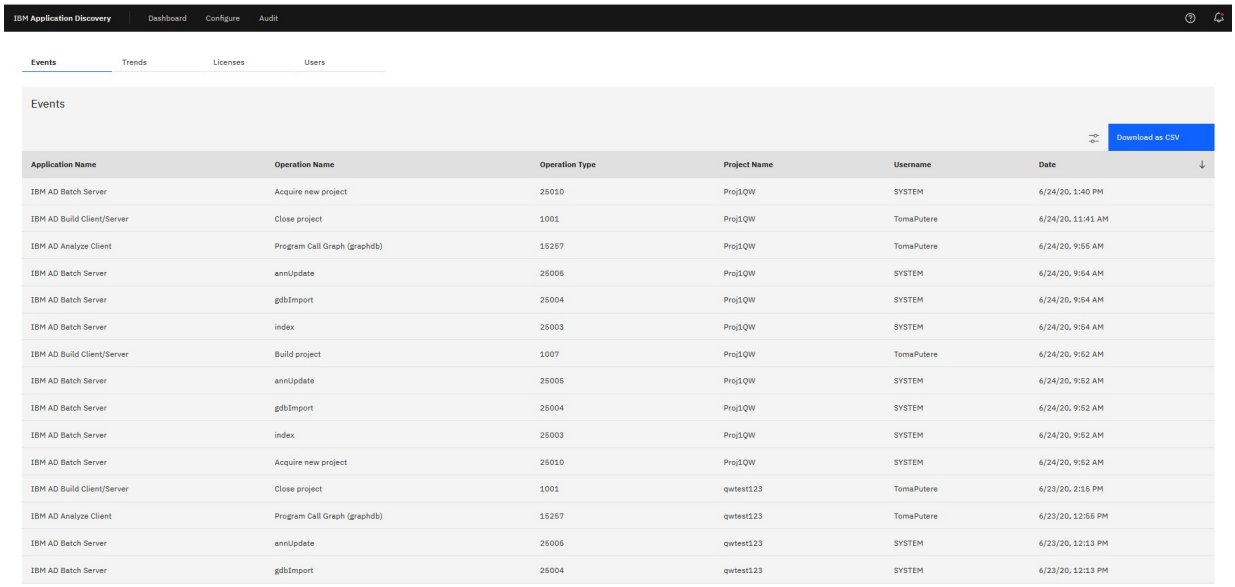
   Note:

   - Wait until the services are restarted, this can take a few minutes to complete.
   - The Audit database will be automatically populated when **IBM Application Discovery Web Service** starts.

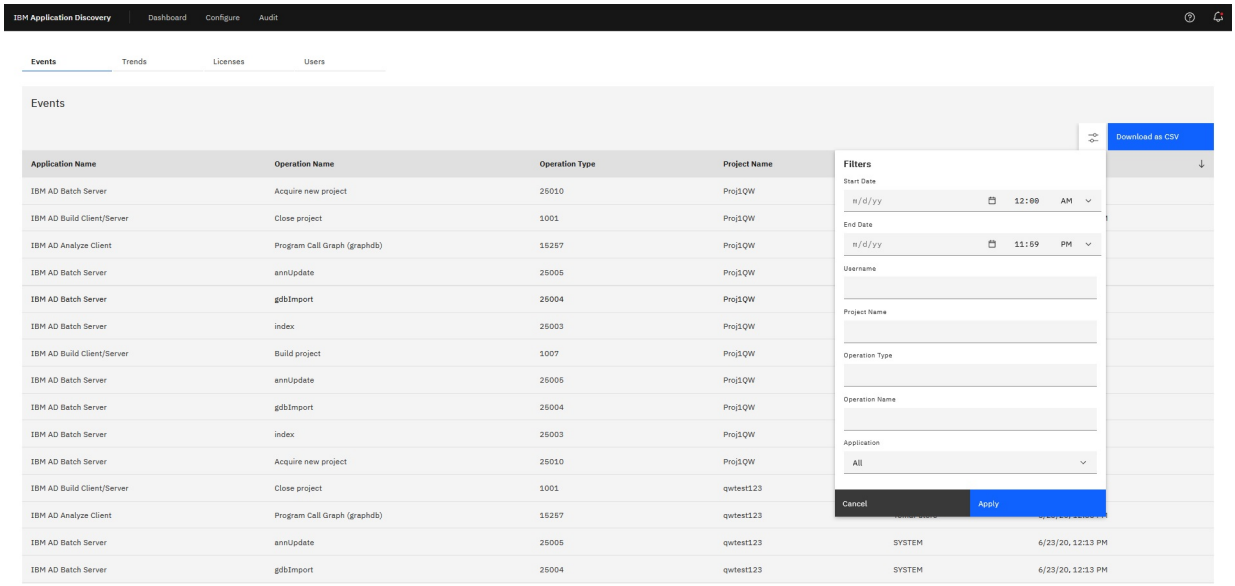
Using the Audit Service Web GUI

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Audit** > **Events**.

   As a result, the **Audit** page is displayed as in the following image.



2. Use filters to search for specific entries by start and end date, username, project name, operation type, operation name, or by application.



3. Alternatively, you can export the results by clicking **Download as CSV**.

```
Application Name, Operation Name, Operation Type, Project Name, Username, Date, IP, Instance Id
IBM AD Build Client/Server,Open Project,1000,SQL_Within,Username,03.03.2020 at 03:00:36,9.228.129.107,null
IBM AD Build Client/Server,Build project,1007,SQL_Within,Username,03.03.2020 at 03:11:05,9.228.129.107,null
IBM AD Build Client/Server,Build project,1007,SQL_Within,Username,03.03.2020 at 03:43:01,9.228.129.107,null
IBM AD Build Client/Server,Open Project,1000,Defect24633DBDFile,Username,03.03.2020 at 03:43:12,9.228.129.107,null
IBM AD Build Client/Server,Close project,1001,SQL_Within,Username,03.03.2020 at 03:43:15,9.228.129.107,null
IBM AD Batch Server,gdbImport,25004,Defect24633DBDFile,SYSTEM,03.03.2020 at 03:57:57,9.228.129.107,4ed6d264-441e-4f8d-bb9d-c8c7302e41fc
IBM AD Batch Server,index,25003,Defect24633DBDFile,SYSTEM,03.03.2020 at 03:57:57,9.228.129.107,4ed6d264-441e-4f8d-bb9d-c8c7302e41fc
IBM AD Batch Server,annUpdate,25005,Defect24633DBDFile,SYSTEM,03.03.2020 at 03:57:57,9.228.129.107,4ed6d264-441e-4f8d-bb9d-c8c7302e41fc
```

Modifying the logging level

You can modify the logging level of the Audit Service by either updating the configuration file or sending a PUT request to the REST API. Accepted logging level values are INFO, WARN, and ERROR. If you modify the logging level at run time, it takes five seconds for the change to take effect.

1. To modify the logging level by updating the configuration file, open *<IBM ADDI Installation Folder>*`\IBM AD Web Services\wlp\usr\servers\ad_server\apps\expanded\ad-audit.war\WEB-INF\classes\log4j2.xml`, find the com.ibm.ad.audit logger, change the value of the `level` attribute, and then save the change.
2. To modify the logging level by sending a PUT request, use the *libertyURL*`/ad-audit/logging/change-level` endpoint, and send the request body with input.

   Note: This is a temporary change, which is lost after the Liberty server restart.

   Example

   ```
   http://localhost:9080/ad-audit/logging/change-level
   ```

   ```
   payload:
   {
       "name":"com.ibm.ad.audit",
       "level":"INFO"
   }
   ```

Modifying the wait time for receiving IBM AD Batch Server status

After an action is finished in IBM AD Build Client, audit events are expected to be sent from IBM AD Batch Server in a given time period. By default, the Audit Service waits for two minutes. If longer wait time is needed, you can follow the steps to modify the value.

1. Stop IBM Application Discovery WebSphere Liberty Profile Service.

   To stop the service, open Windows Services (`services.msc`), locate IBM Application Discovery WebSphere Liberty Profile Service, and then click **Stop**.

2. Open *<IBM ADDI Installation Folder>*\IBM AD Web Services\wlp\usr\servers\ad_server\conf.audit\audit.properties, add the following line, and then save the change.

```
com.ibm.ad.audit.timeout.batch=time_value
```

com.ibm.ad.audit.timeout.batch
Specifies the wait time for receiving IBM AD Batch Server status. Specify the value in milliseconds, for example, 120000, which equals two minutes.

3. Restart IBM Application Discovery WebSphere Liberty Profile Service.

   To restart the service, open Windows Services (services.msc), locate IBM Application Discovery WebSphere Liberty Profile Service, and then click **Start**.

IBM AD BRD

IBM AD BRD allows IBM AD to save data that is used to define Business Rules within IBM ADDI Extension.

Configuring the BRD Service

1. Open the server.xml file from \wlp\usr\servers\ad_server.
2. Enable **IBM AD BRD**, by removing the <!-- --> comment around its <webApplication> element.

```
<webApplication contextRoot="brd-ws" id="com.ibm.ad.brd.restapi"
location="com.ibm.ad.brd.restapi.war" name="ad-brd">
        <classloader privateLibraryRef="config.brd-ws"/>
    </webApplication>
```

3. Go to \wlp\usr\servers\ad_server\conf.brd-ws and in application.properties file, enter the desired values for the properties detailed below.
4. Define the basic authentication username and password for IBM AD BRD Web Service. Once configured, these credentials will be used to create a server connection to IBM AD BRD Server in ADDI Extension.

   Note: The username is case-insensitive while the password is case-sensitive.

```
## Define the basic authentication username for IBM AD BRD Service
## The username is case-insensitive
ad.user=

## Define the basic authentication password for IBM AD BRD Service
## The password is case-sensitive
ad.password=
```

5. Add the IP of the computer where the **IBM AD Configuration Server** host is installed.

```
## IBM AD Configuration Server host
ccs.server.host=
```

6. Add the port number to be used by **IBM AD Configuration Server**; the default port is 2181, if no other value is specified the default value will be used.

```
## IBM AD Configuration Server port
## default 2181
#ccs.server.port=
```

7. Add the ID of the Environment set in **IBM AD Configuration Server**.

```
## IBM AD Configuration Server environment
ccs.environment=
```

8. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

   Note: Wait until the services are restarted, this can take a few minutes to complete.

STEP 3. (Optional) Configuring IBM AD Validation Service

IBM AD Validation Service component is specific only for ChangeMan ZMF users, therefore it is not part of the *must have* components installation.

IBM AD Validation Service is automatically installed during the **IBM AD Build** installation.

IBM AD Validation Service acts like a listener and is linked directly with **IBM AD Connect for Mainframe** component (Mainframe Agents).

After **IBM AD Validation Service** is installed, go to `<IBM AD Build Client installation folder>\Bin\Release\IBMApplicationDiscoveryValidationServer\SampleConf`.

Select all configuration files and copy them to `<IBM AD Build Client installation folder>\Bin\Release\IBMApplicationDiscoveryValidationServer.`

Next, perform the following configurations.

1. Configure `ProjectsMapping.txt` to have a valid input. This is the configuration file for defining the mapping between the projects that are used to download mainframe members, applications, and subsystems.

   Note: Comparing with the `ProjectsMappingParallelBuild.txt` file, the projects that are specified in the `ProjectsMapping.txt` file do not need to contain the virtual folder that is specified in the `FoldersMapping.txt` file, as they are not used for builds. However, a z/OS connection must be attached and configured to the projects.

   Each line of the configuration file must have the following comma-separated values format:

   `<ProjectName>, <ApplicationName>, <Subsystem>`

   Note:

   - `<ProjectName>` represents the project that is defined in **IBM AD Build Client**.
   - `<ApplicationName>` and `<Subsystem>` are defined in **ChangeMan ZMF**.

   Example of the configuration file:

   ```
   Project1, App1, Subsys1
   ```

2. Configure `IncludesOrder.txt` to have a valid input. This is the configuration file for defining the ChangeMan Baseline Libraries Types and the order of COBOL Includes locations. This configuration file is used later on while you set up the path for the COBOL Include folders

   The configuration file must have the following comma-separated values format:

   `<Type1>, <Type2>,…, <Type`*`n`*`>`

   Example of the configuration file:

   ```
   CPY, INC, IND, CPA
   ```

   Note: It is EXTREMELY important to add the types in the order in which the include files must be looked after.

3. Configure `FoldersMapping.txt` to have a valid input. This is the configuration file for defining a mapping between a type of a mainframe member, that is defined in ChangeMan ZMF, and a virtual folder name of an **IBM AD** project. This configuration file is used during the synchronize phase of the validation process.

   Each line of the configuration file must have the following comma-separated values format:

   `<MemberType>, <VirtualFolderName>`

   - `<MemberType>` is defined in **ChangeMan ZMF**.
   - `<VirtualFolderName>` is defined in **IBM AD Build Client**.

   Examples of the configuration file:

```
COB, zOS Cobol
```

```
ASM, Assembler
```

4. Configure `ServicePort.txt` to have a valid input. This is the configuration file for defining the Service's port.

   The configuration file must have the following format:

   `<Port Number>`

   Any available port can be used, for example:

   ```
   48000
   ```

5. Enable or disable sending feedback to the mainframe by configuring the `LoopbackResults.txt` file with one the following values:

   Y
   Enables sending feedback to the mainframe according to the weight of rules.

   N
   Disables sending feedback to the mainframe.

6. Set parallel validation parameters for the maximum-allowed values by configuring the `ParallelValidationParameters.txt`.

   The configuration file must have the following comma-separated values format:

   `<Number_of_validations_in_parallel>,<Number_of_components_per_validation>`

   Note:

   - Do not set the number of validations in parallel greater than the number of CPU cores. Otherwise, the validation process might be unstable.
   - Do not set the number of components per validation greater than 20. Otherwise, the performance might be negatively affected.

   Examples of the configuration file:

   ```
   4,10
   ```

   Allows a maximum of four validation instances in parallel, and a maximum of 10 stages or members that are allocated for each instance. You can set these values for a computer with 4 CPU cores.

   ```
   8,15
   ```

   Allows a maximum of eight validation instances in parallel, and a maximum of 15 members that are allocated for each instance. You can set these values for a computer with 8 CPU cores.

7. Configure the mapping between the projects that are used to compile the members to be validated in parallel, applications, and subsystems. Set the mapping values in the `ProjectsMappingParallelBuild.txt` file.

   Each line of the configuration file must have the following comma-separated values format:

   `<ProjectName>, <ApplicationName>, <Subsystem>`

   Note:

   - `<ProjectName>` represents the project that is defined in **IBM AD Build Client**.
   - `<ApplicationName>` and `<Subsystem>` are defined in **ChangeMan ZMF**.
   - The number of the projects that are mapped to one pair of an application and a subsystem must be greater than or equal to the maximum number of validations in parallel, which is specified in the `ParallelValidationParameters.txt` file. Otherwise, the service cannot start.
   - Comparing with the `ProjectsMapping.txt` file, the projects that are specified in the `ProjectsMappingParallelBuild.txt` file do not need to have a z/OS connection that is

attached and configured, as they are used only for builds. However, the projects must contain the virtual folder that is specified in the `FoldersMapping.txt` file.

The following example shows the mapping configurations for eight validations in parallel:

```
Project1, App1, Subsys1
Project2, App1, Subsys1
Project3, App1, Subsys1
Project4, App1, Subsys1
Project5, App1, Subsys1
Project6, App1, Subsys1
Project7, App1, Subsys1
Project8, App1, Subsys1
Project9, App2, Subsys1
Project10, App2, Subsys1
Project11, App2, Subsys1
Project12, App2, Subsys1
Project13, App2, Subsys1
Project14, App2, Subsys1
Project15, App2, Subsys1
Project16, App2, Subsys1
```

8. Configure the completion code for messages by configuring the `CompletionCodeVsMessage.txt` file.

   Each line of the configuration file must have the following pipe-delimited format:

   `<CompletionCode>|<DescriptiveMessage>`

   Note: The descriptive message must have a maximum length of 23 **characters**.

   Example of the configuration file:

   ```
   0|Validation Success
   4|Validation Warning
   8|Validation Failed
   ```

   Each of the numbers in the example reflects the weight of the rule that is specified in the `ruleBased.properties` file.
9. Configure the approval request parameters in the `ApprovalRequestParameters.txt` file.

   Each line of the configuration file must have the following comma-separated values format:

   `<ProjectName>,<ProcLibrary>`

   Note:

   - `<ProjectName>` represents the project that is defined in **IBM AD Build Client**.
   - `<ProcLibrary>` is a PDS/E library that is defined in **ChangeMan ZMF**.
   - `<ProcLibrary>` must have a maximum length of 23 **characters**.

   Example of the configuration file:

   ```
   Project1,PJ.PROCLIB.S814
   ```

10. Configure the encryption policy by specifying a flag value in the `TLSEncryption.text` file.

    This configuration file contains a single flag byte. The flag can take the value Y or N. It is not case-sensitive. Y indicates that the Application Transparent Transport Layer Security (AT-TLS) encryption is enabled. If N is specified or the flag value is missing, the communication will not be encrypted.

1. Go to `<IBM AD Build Client installation folder\Bin\Release\IBMApplicationDiscoveryValidationServer\ReportsGenerator\sample-conf` folder. This `sample-conf` folder contains the templates for all the configuration files needed to customize the functioning of **ReportsGenerator**. To configure **ReportsGenerator**, copy the templates from the `sample-conf` folder, and place them in the `conf` folder.

   Once this step is completed and before you start **IBM AD Validation Service**, some preliminary configurations are required. You need to specify on which projects you want **ReportsGenerator** to run the reports, which reports to generate, where to store the generated reports etc. The configuration

parameters are stored in `server.properties`, `project.properties`, `ruleBased.properties` and `ruleBasedConfig.properties` files.

For more information, see the following sections that contain a detailed description of the parameters available in these four files:

- “Server Properties” on page 78
- “Global Settings” on page 81
- “ruleBased.properties File” on page 87
- “ruleBasedConfig.properties File” on page 87

Note: ReportsGenerator is invoked by Validation Server after the Build process.

2. Start **IBM AD Validation Service**: Click **Start** and then select **All Programs** > **IBM Application Discovery Build Client** > **Start IBM Application Discovery Validation Service**.

   The service can also be started from Windows Services (`services.msc`) by locating `IBMApplicationDiscoveryValidationServer` and pressing **Start**.

   Important: For monitoring the **Validation Service** tasks, see “Finding Log Files” on page 537.

STEP 4. Configuring Authentication Server (DEX)

The current implementation of the Authentication and Authorization feature, is based on OAuth2 authorization framework, and needs to have available an Authentication Server (DEX). The next configurations are made in the context of using the Authentication Server (DEX), which is provided in the IBM ADDI installer.

Note: These configurations are only made if the authentication of the users, who have access to the resources, is used.

The provided Authentication Server is based on DEX and provides an authentication solution, which connects through LDAP to **Secure Storage**. For more information about DEX, go to the DEX documentation.

Note: DEX can be replaced by any other Authentication Server that supports the OAuth2 protocol.

Follow the configuration steps that are needed to have up and running Authentication Server (DEX):

1. Configure the parameters that are present in the `dex.yaml` file
2. Make Authentication Server (DEX) available in IBM AD Configuration Server
3. Restart Authentication Server (DEX)

1. Configure the parameters that are present in the `dex.yaml` file

On the machine where Authentication Server is installed, go to `<IBM ADDI Installation Folder>/Authentication Server (DEX)/sample-conf/` and copy the `dex.yaml` file to `<IBM ADDI Installation Folder>/Authentication Server (DEX)/conf/`. Open the `dex.yaml` file by using a text editor and enter the desired values for the properties that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Set the **`issuer`** parameter as follows.

   a. If the communication to and from DEX is done through https, the **`issuer`** parameter has the following format:

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
https://<machine name where DEX is installed>.<machine domain>:<port>/dex
```

Example:

```
issuer: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

b. If the communication to and from DEX is done through http, the **issuer** parameter has the following format:

```
http://<machine name where DEX is installed>.<machine domain>:<port>/dex
```

Example:

```
issuer: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

2. The next section can be configured as follows:

a. If the communication to and from DEX is done through https, generate the TLS certificates for Authentication Server (DEX) and add the paths for the `certificate(.crt)` and the `key(.key)` files in the TLS`Cert` and TLS`Key` fields. The default port is 7600.

```
storage:
  type: sqlite3
  config:
    file: dex.db
frontend:
  theme: addi
web:
  https: 0.0.0.0:7600
  TLSCert: C:\certs\dex.crt
  TLSKey: C:\certs\dex.key
```

b. If the communication to and from DEX is done through http, comment the TLS`Cert` and TLS`Key` fields.

```
storage:
  type: sqlite3
  config:
    file: dex.db
frontend:
  theme: addi
web:
  http: 0.0.0.0:7600
  #TLSCert:
  #TLSKey:
```

3. The **skipApprovalScreen** parameter can be set to `true` or `false`. The `true` value offers the possibility to skip the **"Grant access screen"** after the user logs in.

```
oauth2:
  skipApprovalScreen: true
```

4. Starting with IBM AD V5.1.0.9, DEX is upgraded to version 2.25.0. With this version of DEX, you can run requests to services that can be called by URLs with other tools, for example, Postman. To run such requests, you must add and configure the **passwordConnector** parameter under `oauth2`. The value of this parameter must be the connector ID that is defined by the **id** parameter in the **connectors** section.

```
oauth2:
  skipApprovalScreen: true
  passwordConnector: connector_id
```

Note: Besides configuring the **passwordConnector** parameter, a security token is needed to complete the requests. You can generate a token with any tool that can be used to run API URLs. The following steps show how to generate and use a token with Postman for the requests.

a. Go to the **Authorization** tab in Postman, and then select **OAuth 2.0** from the **TYPE** drop-down list.

b. Complete the settings on the right.

Token Name
Enter the name that you want to use for the token.

Grant Type
Select **Password Credentials**.

Access Token URL
Enter the authentication issuer URL of the DEX authentication server, which can be found in the `dex.yaml` file. This URL is to exchange an authorization code for an access token.

Client ID
Enter the IBM AD Analyze Client application ID that is defined by the **id** parameter in the **StaticClients** section in file `dex.yaml`.

Client Secret
Enter the client secret that is defined by the **secret** parameter in the **StaticClients** section in file `dex.yaml`.

Username
Enter the user name of any user that can access projects and conduct analysis in IBM AD Analyze Client.

Password
Enter the password of the preceding user.

Scope
Enter the scope of access that you request, which might include multiple space-separated values. The following scope is used in IBM AD Analyze Client:

```
offline_access openid profile email groups
```

Client Authentication
Select **Send as Basic Auth header**.

c. Click **Get new Access Token** > **Proceed**. The generated token can be found in the **id_token** field in the pop-up **Manage Access Tokens** window.

d. To use the generated token, go to the **Authorization** tab, and then select **Bearer Token** from the **TYPE** drop-down list. Copy and paste your token in the **Token** field.

5. The **connectors** section can be configured as follows:

a. Set the **host** parameter, including the default port 389 or 636.

The **host** parameter has the following format:

```
host: << IP:PORT >>
```

Example:

```
config:
    host: WIN-NSSMI7A1KJQ.ferdinand2.com:636
```

b. The **InsecureNoSSL** parameter can be set as follows:

- If the **host** parameter was set using the default port 389, set the **InsecureNoSSL** to `true`.

```
insecureNoSSL: true
```

- If the **host** parameter was set using the default port 636, set the **InsecureNoSSL** to `false`.

```
insecureNoSSL: false
```

c. Set the **bindDN** parameter by adding the account that has the rights for the LDAP bind action.

Note: To add the account that has rights for LDAP bind action run `adsiedit.msc` on the Active Directory machine and load the current domain. Right click on *CN=Users* and *CN=Administrator*, select *Properties* and search for *distinguishedName* attribute. For more information, see ADSI Edit (adsiedit.msc).

```
# This would normally be a read-only user.
    bindDN: CN=Administrator,CN=Users,DC=ferdinand2,DC=com
```

d. Set the **bindPW** parameter by adding the account's password that has the rights for the LDAP bind action.

```
bindPW: password
```

e. Do not modify the value of the **usernamePrompt** parameter.

```
usernamePrompt: email address
```

6. Under the **userSearch** section, only modify the value of the **baseDN** parameter:

Note: The **baseDN** parameter contains the base location of all User Accounts.

```
userSearch:
      baseDN: dc=ferdinand2,dc=com
      filter: "(objectClass=person)"
      username: userPrincipalName
      # "DN" (case sensitive) is a special attribute name. It indicates that
      # this value should be taken from the entity's DN not an attribute on
      # the entity.
      idAttr: DN
      emailAttr: userPrincipalName
      nameAttr: cn
```

7. Under the **groupSearch** section, only modify the value of the **baseDN** parameter:

Note: The **baseDN** parameter contains the base distinguished name of the groups in LDAP registry.

```
groupSearch:
      baseDN: cn=Users,dc=alpaca,dc=com
      filter: "(objectClass=group)"

      # A user is a member of a group when their DN matches
      # the value of a "member" attribute on the group entity.
      userAttr: DN
      groupAttr: member

      # The group name should be the "cn" value.
      nameAttr: cn
```

8. The **StaticClients** section, can be configured as follows:

```
staticClients:
- id: analyze-client
  redirectURIs:
  - 'http://127.0.0.1:9999/callback'
  name: 'AD Client'
  secret: 38fcbc1a-3a65-11e9-b210-d663bd873d93
```

Where:

- **id** is the generic name that is given for IBM AD Analyze Client.
- **redirectURLs** takes as value the localhost IP and a generic port that is used for callback to Authentication Server (DEX).
- **name** takes as value "AD Client".
- **secret** is a secret that is shared among application.

Configuration Examples:

- When the communication to and from DEX is done through https the `dex.yaml` file is configured as follows:

```
issuer: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
storage:
  type: sqlite3
  config:
    file: dex.db

frontend:
  theme: addi

web:
  https: 0.0.0.0:7600
  TLSCert: C:\certs\dex.crt
  TLSKey: C:\certs\dex.key

oauth2:
  skipApprovalScreen: true

connectors:
- type: ldap
  name: ADLDAP
  id: ldap
  config:
    host: WIN-NSSMI7A1KJQ.ferdinand2.com:636


    # No TLS for this setup.
    insecureNoSSL: false

    # This would normally be a read-only user.
    bindDN: CN=Administrator,CN=Users,DC=ferdinand2,DC=com
    bindPW: Admin15_

    usernamePrompt: email address

    userSearch:
      baseDN: dc=ferdinand2,dc=com
      filter: "(objectClass=person)"
      username: userPrincipalName
      # "DN" (case sensitive) is a special attribute name. It indicates that
      # this value should be taken from the entity's DN not an attribute on
      # the entity.
      idAttr: DN
      emailAttr: userPrincipalName
      nameAttr: cn

    groupSearch:
      baseDN: dc=ferdinand2,dc=com
      filter: "(objectClass=group)"
      # A user is a member of a group when their DN matches
      # the value of a "member" attribute on the group entity.
      userAttr: DN
      groupAttr: member

      # The group name should be the "cn" value.
      nameAttr: cn

staticClients:
- id: analyze-client
  redirectURIs:
  - 'http://127.0.0.1:9999/callback'
  name: 'AD Client'
  secret: 38fcbc1a-3a65-11e9-b210-d663bd873d93
```

- When the communication to and from DEX is done through http the `dex.yaml` file is configured as follows:

```
issuer: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
storage:
  type: sqlite3
  config:
    file: dex.db

frontend:
  theme: addi

web:
```

```
  http: 0.0.0.0:7600
  #TLSCert:
  #TLSKey:

oauth2:
  skipApprovalScreen: true

connectors:
- type: ldap
  name: ADLDAP
  id: ldap
  config:
    host: WIN-NSSMI7A1KJQ.ferdinand2.com:389


    # No TLS for this setup.
    insecureNoSSL: true

    # This would normally be a read-only user.
    bindDN: CN=Administrator,CN=Users,DC=ferdinand2,DC=com
    bindPW: Admin15_

    usernamePrompt: email address

    userSearch:
      baseDN: dc=ferdinand2,dc=com
      filter: "(objectClass=person)"
      username: userPrincipalName
      # "DN" (case sensitive) is a special attribute name. It indicates that
      # this value should be taken from the entity's DN not an attribute on
      # the entity.
      idAttr: DN
      emailAttr: userPrincipalName
      nameAttr: cn

    groupSearch:
      baseDN: dc=ferdinand2,dc=com
      filter: "(objectClass=group)"
      # A user is a member of a group when their DN matches
      # the value of a "member" attribute on the group entity.
      userAttr: DN
      groupAttr: member

      # The group name should be the "cn" value.
      nameAttr: cn

staticClients:
- id: analyze-client
  redirectURIs:
  - 'http://127.0.0.1:9999/callback'
  name: 'AD Client'
  secret: 38fcbc1a-3a65-11e9-b210-d663bd873d93
```

2. Make Authentication Server (DEX) available in IBM AD Configuration Server

The authentication process is conducted through DEX or through any other Authentication Server that supports the OAuth2 protocol. Authentication Server (DEX), which is delivered through the IBM ADDI installer, is an identity service that uses OpenID Connect and it is used in the following configurations.

Go to IBM AD Configuration Server and make Authentication Server (DEX) available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers** > **Authentication Server (DEX)**. The **Authentication Server (DEX)** settings page is displayed.
2. Enter the following information:

   Application ID
   Expects the generic name that is given for IBM AD Analyze Client.

   Secret
   Expects a secret that is shared between Authentication Server (DEX) and IBM AD Analyze Client.

Callback URL
Expects a URL composed by localhost IP and a generic port that is used for callback to Authentication Server (DEX).

Timeout (in seconds)
Expects the time for the client to respond.

Authentication server host
Expects the authentication server host that is used by Authentication Server (DEX).

Authentication discovery path
Expects the authentication discovery path that is used by Authentication Server (DEX).

3. If you want to use Authentication Server (DEX), make sure that the **Enable Authentication** check box is selected.
4. Click **Save**.

3. Restart Authentication Server (DEX)

Once the configuration is done, go to the **Dashboard** tab, in IBM® AD Configuration Server, click the menu button of **Authentication Server (DEX)**, and select **Restart Service**.

Note: Wait until the service is restarted, this can take a few minutes to complete.

If the service does not start, check the `dex.log` file under `<IBM ADDI Installation Folder>/Authentication Server/` folder.

STEP 5. Configuring IBM AD File Service

Follow the configuration steps that are needed to have up and running **IBM AD File Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD File Service available in IBM AD Configuration Server
3. Restart IBM AD File Service

1. Configure the parameters that are present in the `conf.yaml` file

On the machine where **IBM AD File Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Set the **https** parameter as follows:
   a. If the **https** parameter is set to *false*, a non-secured communication is used.

   ```
   #if true, tls information (key, cert) must be specified
   https: false
   ```

   b. If the **https** parameter is set to *true*, a secured communication is used.

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
#if true, tls information (key, cert) must be specified
https: true
```

2. If the **https** parameter is set to *true* and the TLS certificate for **IBM AD File Service** are generated, enter the paths of the `certificate(.crt)` and the `key(.key)` files. If the **`https`** parameter is set to *false* leave blank the following lines. Example:

```
#mandatory if https: true
tls:
    key: C:\certs\file.service.key
    cert: C:\certs\file.service.crt
```

3. Set the **authSrv** parameter as follows:

   a. If the value of the **https** parameter is set to *true*, add the URL of **Authentication Server (DEX)** where **authSrv** parameter is present. **Authentication Sever (DEX)** that belongs to the **IBM AD** package is used. For more information, see "STEP 4. Configuring Authentication Server (DEX)" on page 111. Example:

```
#authentication server URL
authSrv: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

   b. If the value of the **https** parameter is set to *false* and the **Authorization and Authentication** feature is *enabled*, add the URL of **Authentication Server (DEX)**. Example:

```
#authentication server URL
authSrv: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

4. The **caseSensitive** parameter can be set to *true* or *false*. Through this parameter, you set the mapping type (case-sensitive) of the folders under which the resources are located.

```
caseSensitive: false
```

5. Set the **disableAuth** parameter to *false*. The *false* value keeps enabled the authentication.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: false
```

6. If DEX is used, the default value of the matchUsers parameter is false, which means that the matching is made by **group names**. You only need to ensure that the groups names are added in the Configuration Dashboard and the group names in the Configuration Dashboard match the group names that are defined in LDAP. Then, all the LDAP users that are contained in those groups will have the access in the Analyze Client.

```
matchUsers: false
```

   If DEX is not used, and you want to authorize users but not groups that exist in the Active Directory, set the matchUsers parameter to be true. Then, you can manually add individual users in the Configuration Dashboard and allocate the added users to a group with a name that is wanted by the administrator, and afterwards allocate the added users to a group that has access to some projects. For more information, see "Adding a User" on page 49.

```
matchUsers: true
```

   Note: It is not recommended to import the list of users in **IBM AD Configuration Server** because it might be time consuming to keep the list synchronized with the one from **Secure Storage**.

7. To use groups in Application Discovery when DEX is used, you must ensure that a perfect match exists to the groups in the Active Directory, which means that the group names in Application Discovery are case-sensitive with respect to the corresponding LDAP group names. If a user is added to a group in

the Dashboard, the same group name also must exist in Active Directory and contain the user so that the user can be authenticated and authorized in the Analyze Client.

2. Make IBM AD File Service available in IBM AD Configuration Server

The access rights of users or users' groups are mapped to a certain folder that contains the source files. A user can start the analysis on the source files if it has read access rights.

After **IBM AD File Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD File Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **File Service**. The **File Service** settings page is displayed.
2. In the **Mappings** section, click the **Add** button to add a new mapping and enter the following information:
   - **Remote** - add the remote path that can be used to query this service, for example, a UNC path or a local path where the resources/projects reside.
   - **Local** - add the local path that mirrors the remote path. If missing, it is identical to **Remote**.

     Note:

     – In case you have multiple physical folders that reside resources or projects, an entry for each folder needs to be added in the **mapping** section.

     – In order to see **Flow Chart** analysis in **IBM AD Analyze Client** for a specific project, you need to add the path of the project folder or the path of the folder that contains all projects.

       Examples:

       - `\\9.20.128.222\Projects` - path to all projects.
       - `\\9.20.128.222\ADProject` - path to a specific project.
   - Define which **User Groups** can access the related mapping folders.

     Note: If no **User Groups** are defined, all users will have access to the related mapping folders.
   - Click **Save**.
3. In the **Group Settings** section, enter the following information:
   - **Session Timeout** - add the session timeout to determine how long the service needs to remain connected to **IBM AD Configuration Server** before it must perform a new connection. The value format must be a whole number followed by 's' for seconds, 'm' for minutes or 'h' for hours.
   - **Groups Polling** - add the refresh time to check periodically the groups that are present in **IBM AD Configuration Server**. The value format must be a whole number followed by 's' for seconds, 'm' for minutes or 'h' for hours.
4. Click **Save**.

3. Restart IBM AD File Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **File Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. If the service does not start, check the `.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/` folder and locate the `micro-srcd-`$x.x.x$`.bin` file.

2. In case that the `.bin` file is not executable, open a terminal and run the following command for flagging them as executable:

```
chmod +x micro-srcd-x.x.x.bin
```

3. If the service does not start, check the `.log` file under <IBM ADDI Installation Folder>/IBM Application Discovery File Service/ folder.

STEP 6. Configuring IBM AD Manual Resolutions Service

Follow the configuration steps that are needed to have up and running **IBM AD Manual Resolutions Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Manual Resolutions Service available in IBM AD Configuration Server
3. Restart IBM AD Manual Resolutions Service

1. Configure the parameters that are present in the `conf.yaml` file

On the machine where **IBM AD Manual Resolutions Service** is installed, go to <IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/conf/ and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the /conf folder, go to <IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/sample-conf/ and copy the `conf.yaml` file in the /conf folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Add the port of **IBM AD Configuration Server**.

```
## Coordination and Configuration Server port
## default 2181
ccs.server.port: 2181
```

2. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

```
#if communication should be secured with TLS
https: false
```

   b. If the **https** parameter is set to *true*, a secured communication is used.

   Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
#if communication should be secured with TLS
https: true
```

   Note: If the **https** parameter is set to *true*, an additional step needs to be performed.
   Locate `startServer.bat` file under <IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/ and replace the following line:

```
set tlsoptions=
```

   with:

```
SET keystorepath=<"path_to_keystore">
SET keystorepass=<"password_of_keystore">
set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
-Djavax.net.ssl.keyStorePassword="%keystorepass%"
```

Where:

- **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Manual Resolutions Service**.
- **Keystore password** is the keystore password.

3. Set the **authSrv** parameter as follows:
   a. If the value of the **https** parameter is set to *true*, add the URL of **Authentication Server (DEX)** where **authSrv** parameter is present. **Authentication Sever (DEX)** that belongs to the **IBM AD** package is used. For more information, see "STEP 4. Configuring Authentication Server (DEX)" on page 111. Example:

   ```
   #authentication server URL
   authSrv: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
   ```

   b. If the value of the **https** parameter is set to *false* and the **Authorization and Authentication** feature is *enabled*, add the URL of **Authentication Server (DEX)**. Example:

   ```
   #authentication server URL
   authSrv: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
   ```

4. Set the **disableAuth** parameter to *false*. The *false* value keeps enabled the authentication.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: false
```

2. Make IBM AD Manual Resolutions Service available in IBM AD Configuration Server

The **IBM AD Manual Resolutions Service** manages the manually added resolutions and it is mandatory to be configured if you want to use Callgraph-based analyzes (graphs or reports).

After **IBM AD Manual Resolutions Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Manual Resolutions Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Manual Resolutions Service**. The **Manual Resolutions Service** settings page is displayed.
2. If necessary, you can modify the path where the **journal files** are created. The path where these files are generated is separated from the project's path and needs to be accessible only for **IBM AD Manual Resolutions Service**. Once a project is imported, a folder with the same name is generated in the related path and it hosts all the files that are needed to manage dynamic call resolutions. Example:

   ```
   C:\AD\Resolutions
   ```

3. Optionally, the main path where the manual resolutions are created for each project, can be overwritten. Click **Add**, select the target project, and add the path where the manual resolutions are created for the target project.
4. Click **Save**.

3. Restart IBM AD Manual Resolutions Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Manual Resolutions Service**, and select **Restart Service**.

Note: Wait until the service is restarted, this can take a few minutes to complete.

2. If the service does not start, check the `manualres.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/log` folder.

- **On Linux**

1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/` and locate the `startServer.sh` file.
2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

```
chmod +x startServer.sh
```

3. If the service does not start, check the `manualres.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/log` folder.

STEP 7. Configuring IBM AD Mainframe Projects Service

Follow the configuration steps that are needed to have up and running **IBM AD Mainframe Projects Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Mainframe Projects Service available in IBM AD Configuration Server
3. Restart IBM AD Mainframe Projects Service

Note: When configuring **IBM AD Mainframe Projects Service**, you need to add in **IBM AD Configuration Server** the **group names** that are defined in **Secure Storage** to decide which projects are accessible to specific users.

1. Configure the parameters that are present in the `conf.yaml` file

On the machine where **IBM AD Mainframe Projects Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Add the port of **IBM AD Configuration Server**.

```
## Coordination and Configuration Server port
## default 2181
ccs.server.port: 2181
```

2. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

```
#if communication should be secured with TLS
https: false
```

   b. If the **https** parameter is set to *true*, a secured communication is used.

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
#if communication should be secured with TLS
https: true
```

Note: If the **https** parameter is set to *true*, an additional step needs to be performed.
Locate `startServer.bat` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/` and replace the following line:

```
set tlsoptions=
```

with:

```
SET keystorepath=<"path_to_keystore">
SET keystorepass=<"password_of_keystore">
set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
-Djavax.net.ssl.keyStorePassword="%keystorepass%"
```

Where:

- **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Mainframe Projects Service**.
- **Keystore password** is the keystore password.

3. Set the **disableAuth** parameter to *false*. The *false* value keeps enabled the authentication.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: false
```

4. Set the **authSrv** parameter as follows:
   a. If the value of the **https** parameter is set to *true*, add the URL of **Authentication Server (DEX)** where **authSrv** parameter is present. **Authentication Sever (DEX)** that belongs to the **IBM AD** package is used. For more information, see “STEP 4. Configuring Authentication Server (DEX)” on page 111. Example:

```
#authentication server URL
authSrv: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

   b. If the value of the **https** parameter is set to *false* and the **Authorization and Authentication** feature is *enabled*, add the URL of **Authentication Server (DEX)**. Example:

```
#authentication server URL
authSrv: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

2. Make IBM AD Mainframe Projects Service available in IBM AD Configuration Server

Through **IBM AD Mainframe Projects Service** a user is authorized to access the AD projects.

After **IBM AD Mainframe Projects Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Mainframe Projects Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Mainframe Projects Service**. The **IBM AD Mainframe Projects Service** settings page is displayed.
2. Set the **Poll interval** parameter to determine how frequently the projects are tested (seconds).
3. Click **Save**.

3. Restart IBM AD Mainframe Projects Service

- **On Windows**

  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Mainframe Projects Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.

  2. If the service does not start, check the `mfprojs.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/log` folder.

- **On Linux**

  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/` and locate the `startServer.sh` file.
  2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x startServer.sh
     ```

  3. If the service does not start, check the `mfprojs.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/log` folder.

STEP 8. Configuring IBM AD Batch Server

Before running **IBM AD Batch Server**, some preliminary configurations must be performed. You need to specify on which projects you want **IBM AD Batch Server** to run the reports, which reports to generate, where to store the generated reports, and so on. Also, you need to specify the parameters for **IBM AD Web Service**.

The configuration parameters are stored in `server.properties` and `project.properties` files, which can be found in the configuration folder.

Below are the instructions on how to perform a minimal configuration in order to have source code analysis in **IBM AD Analyze Client**. For detailed instructions on how to configure **IBM AD Batch Server**, see `IBM AD Batch Server User Guide`.

Note: Under Linux, in case `.sh` files are not executable, navigate to their installation directory, open a terminal and run the following command for flagging them as executable:

```
chmod +x filename.sh
```

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers** > **Graph Database Server**. The **Graph Database Server** settings page is displayed.

   a. During the installation process through **IBM AD Configuration Wizard**, the password for GraphDB Server's root user was set. If necessary, set a new password and confirm it.

   b. Click **Save**.

2. In the context of **Graph Database**, **OrientDB SSL** can be configured as follows:

   Note: For more information about OrientDB SSL configuration, see Using SSL with OrientDB.

   a. Copy the path to the keystore file that was uploaded to the AD Configuration Service Admin, for example, `<IBM ADDI Installation Folder>\security\<environment-id>\server_keystore.jks`.

   b. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\orientdb\orientdb-community\config`, open the `orientdb-server-`

config.xml, set **OrientDB SSL** to true and update the password for both keystore and truststore as follows:

```
<parameter name="network.ssl.clientAuth" value="true"/>
<parameter name="network.ssl.keyStore" value="<IBM ADDI Installation
Folder>\security\<environment-id>\server_keystore.jks"/>
<parameter name="network.ssl.keyStorePassword" value="password"/>
<parameter name="network.ssl.trustStore" value="<IBM ADDI Installation
Folder>\security\<environment-id>\server_keystore.jks"/>
<parameter name="network.ssl.trustStorePassword" value="password"/>
```

Note: You can enter an encrypted **Keystore Password**. For more information, see "Obtaining an encrypted keystore password" on page 209.

c. In the listeners section of the orientdb-server-config.xml file, update the first line that includes the socket and the default port range for **OrientDB SSL** as follows:

```
<listener protocol="binary" socket="ssl" port-range="2424" ip-address="0.0.0.0"/>
```

d. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\orientdb\orientdb-community\bin, open shutdown.bat, find the line that starts with set JAVA_OPTS, and update the password for both keystore and truststore as follows:

```
set JAVA_OPTS=-Djava.awt.headless=true -Dclient.ssl.enabled=true
-Djavax.net.ssl.keyStore="<IBM ADDI Installation Folder>\security\<environment-
id>\server_keystore.jks" -Djavax.net.ssl.keyStorePassword=password
-Djavax.net.ssl.trustStore="<IBM ADDI Installation Folder>\security\<environment-
id>\server_keystore.jks" -Djavax.net.ssl.trustStorePassword=password
```

Note: Enter the same password that was entered in step "2.b" on page 124.

e. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\, open startServer.bat and set the **gdbOptions** parameter as follows:

```
set gdbOptions=-DgdbImport.client.ssl.enabled=true -DgdbImport.javax.net.ssl.trustStore=<<IBM ADDI Installation
Folder>\security\<environment-id>\server_keystore.jks> -DgdbImport.javax.net.ssl.trustStorePassword=password
-DgdbImport.javax.net.ssl.keyStore=<<IBM ADDI Installation Folder>\security\<environment-id>\server_keystore.jks>
-DgdbImport.javax.net.ssl.keyStorePassword=password
```

Note: At this point, **OrientDB SSL** is configured to run with the **root** username and the password that is configured above.

3. The project.properties file contains a set of global settings, followed by the specific settings for each type of component. The global settings specify the projects on which the **IBM AD Batch Server** will operate and which components will run on the specified projects.

   Detailed information on the additional components that can be considered for this service.

Table 6. Optional Components

Component	Description
ruleBased	The **Rule Based** component generates reports for the resources specified in the configuration files according to the rules and parameters defined in the corresponding configuration files. **Note:** If the **Rule Based** component is used, make sure that the ruleBased.properties file is configured. For more information, go to "ruleBased.properties File" on page 87.
reports	The **Reports** component is used to generate the complexity reports. For more information, go to "Complexity Reports" on page 359.
cobolPP	The **cobolPP**, **jclPP**, and **pl1PP** components generate the expanded sources for Cobol, JCL, and PL/I. For more information, go to "View Expanded Source" on page 318.
jclPP	
pl1PP	

Table 6. Optional Components (continued)	
Component	**Description**
wsmetrics	The **wsmetrics** component is needed only if IBM ADDI Extension is used on the system. **Note:** Additionally, in order for the **wsmetrics** component to be executed, make sure that the **gdbImport** component is included in the components list of this service.
adidx	The **addix** component is needed only if IBM ADDI Extension is used on the system. The ADI Index component will index the resources of a project so that the Business Rules Discovery (BRD) feature can display code snippets.

4. On **Linux** only, Open the `mount.properties` file, located under `<installation folder>\IBM Application Discovery Batch Server/conf` folder and specify how the windows shared folders are mounted on the local files system, using the following pattern:

   ```
   \\\\machine IP\\WindowsSharedFolder=/home/user/LinuxFolder
   ```

   Example:

   ```
   \\\\192.168.56.57\\ProjectsSharedPathWindows=/home/user/ ProjectsSharedPathLinux
   ```

5. Optional step: for integration with ADI only, please follow this procedure to set up the **AD Batch Server Web Service**:

   a. Go to *<IBM ADDI Installation Folder>*`\IBM Application Discovery Batch Server\`, and run `authConfigTool.bat` on **Windows** or `authConfigTool.sh` on **Linux**. A command prompt dialog window is displayed. Follow the directions and enter the username and the password that are used by the Web Service then press ENTER. `AuthConfigTool.bat` sets the user and password for Web Service basic access authentication.

   b. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\conf` folder, locate the `webService.properties` file and set the **keystore-file** parameter to enable an encrypted communication. Example:

   ```
   ## ssl keystore file
   keystore-file=keystore.jks
   ```

   Note: The `keystore` file needs to be added in the same `\conf` folder where `webService.properties` is present.

- Restart **Graph Database Server**.
  - Under Windows: Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Graph Database Server**, and select **Restart Service**.

    Note: Wait until the service is restarted, this can take a few minutes to complete.
  - Under Linux: Go to *<IBM ADDI Installation Folder>*`\IBM Application Discovery Batch Server\orientdb\orientdb-community-2.1.25-ibm1\bin\` and run `server.sh`. Make sure this process remains alive.
- Restart **Batch Server Service**.
  - Under Windows: Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Batch Server Service**, and select **Restart Service**.

    Note: Wait until the service is restarted, this can take a few minutes to complete.
  - Under Linux: Go to *<IBM ADDI Installation Folder>*`\IBM Application Discovery Batch Server\`, and run `StartServer.sh`. Make sure this process remains alive.
- (Only in case step 5 from above has been taken) Start **IBM AD Web Service**.

– Under Windows: Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Web Service**, and select **Restart Service**.

  Note: Wait until the service is restarted, this can take a few minutes to complete.

– Under Linux: Go to *<IBM ADDI Installation Folder>*\IBM Application Discovery Batch Server\, and run `startWBServer.sh`. Make sure this process remains alive.

Note: Make sure to restart **IBM AD Batch Server** after modifying the configuration files.

Important: For monitoring the **IBM AD Batch Server** tasks, see "Finding Log Files" on page 537.

Step 8.1 Advanced Configurations for IBM AD Batch Server

After **IBM Application Discovery for IBM Z (AD) Batch Server** is installed, a folder named `sample-conf` is created in the **IBM AD Batch Server** installation folder. This `sample-conf` folder contains the templates for all the configuration files needed to customize the functioning of **IBM AD Batch Server**. To configure **IBM AD Batch Server**, copy the templates from the `sample-conf` folder, and place them in the `conf` folder.

Once the setup is completed, before running **IBM AD Batch Server**, some preliminary configurations should be performed. You need to specify on which projects you want **IBM AD Batch Server** to run the reports, which reports to generate, where to store the generated reports etc. Also, you need to specify the parameters for **IBM AD Web Service**.

The configuration parameters are stored in `server.properties`, `project.properties`, and `webservice.properties` files. The following sections contain a detailed description of the parameters available in these three files as well as a description of other configuration files for specific components such as **Annotations**, **GraphDB**, **Search Index**, **Rule Based**, **Reports**, and **Web Service Metrics**.

Important: It is vital that you set the parameters for the **Search Index**, **Annotations** and **GraphDB** components. Failing to do so will make some of the analyses impossible to generate in **IBM AD Analyze**.

Server Properties

In `server.properties` file, enter the desired values for the properties that are detailed below.

Note: Make sure to restart **IBM AD Batch Server** after you modify the configuration file.

1. Add the IP of the computer where the **IBM AD Configuration Server** host is installed.

```
## Coordination and Configuration Server host
ccs.server.host=127.0.0.1
```

2. Add the port number to be used by **IBM AD Configuration Server**; the default port is 2181, if no other value is specified the default value is used.

```
## Coordination and Configuration Server port
## default 2181
#ccs.server.port=2181
```

3. Add the ID of the Environment set in **IBM AD Configuration Server**.

```
## Coordination and Configuration Server environment
ccs.environment=fb9107c2-8770-43ef-990c-15b387f19698
```

4. By default the audit of **IBM AD Batch Server** components is enabled, by setting it to false the Audit stops.

```
##Audit server
# default true
#audit.enable=true
```

5. Add the IP used by **IBM AD Batch Server** to communicate with **IBM AD Configuration Server**

```
## client host
## default: one of the available ip-s
# client.host=
```

6. Add the IP of the computer where **IBM AD Batch Server** is installed.

```
##aplication server host
#application.server.host=
```

7. Add the port number where **IBM AD Batch Server** listens to admin requests; the default port is 8686, if no other value is specified the default value is used.

```
## port on which the server listens to admin requests
## default: 8686
#application.server.port=
```

8. Specify how much to wait before the server stops (milliseconds); the default value is 5000, if no other value is specified the default value is used.

```
## how much to wait for the server to stop (milliseconds)
## default: 5000
#stop.timeout=
```

9. Specify how frequently database version changes are tested (seconds); the default value is 120, if no other value is specified the default value is used.

```
## how frequently database version changes are tested (seconds)
## default: 120
#poll.interval=
```

10. Specify the installation directory or the place where **IBM AD Batch Server** is installed. All paths are either absolute or relative to the installation directory. Relative paths start with the first directory under `${install.dir}`.

```
## ${install.dir} place where server is installed
## all paths are either absolute or relative to ${install.dir}
## relative paths start with the first directory under ${install.dir}
```

a) Enter the path to the directory where configuration files are stored; the default path is `${install.dir}/conf`; if no other path is specified the default path is used.

```
## path to the directory where configuration files are placed
## default: ${install.dir}/conf
conf.dir={install.dir}/conf
```

Important:

- Under Windows, make sure that all backslash characters in all paths are doubled as in ****. Example:

```
conf.dir=C:\\Program\Files\\IBM\Application\Discovery\Batch\Server/conf
```

- Under Linux, remember that all paths use **/** as separators and spaces are escaped by using ****. Example:

```
conf.dir=/home/username/IBM\Application\Discovery\Batch\Server/conf
```

b) Enter the path to the directory where data files are stored; the default path is `${install.dir}/data`; if no other path is specified the default path is used.

```
## path to the directory where data files are placed
## default: ${install.dir}/data
data.dir={install.dir}/data
```

Important:

- Under Windows, make sure that all backslash characters in all paths are doubled as in ****. Example:

```
data.dir=C:\\Program\Files\\IBM\Application\Discovery\Batch\Server/data
```

- Under Linux, remember that all paths use **/** as separators and spaces are escaped by using ****. Example:

```
data.dir=/home/username/IBM\Application\Discovery\Batch\Server/data
```

c) Enter the path to the directory where temporary data is stored; the default path is `${data.dir}/tmp`; if no other path is specified the default path is used. Example:

```
## path to the directory where temporary data is placed
## default: ${data.dir}/tmp
tmp.dir={data.dir}/tmp
```

Important:

- Under Windows, make sure that all backslash characters in all paths are doubled as in ****. Example:

```
tmp.dir=C:\\Program\Files\\IBM\Application\Discovery\Batch\Server/tmp
```

- Under Linux, remember that all paths use **/** as separators and spaces are escaped by using ****. Example:

```
tmp.dir=/home/username/IBM\Application\Discovery\Batch\Server/tmp
```

Project Properties

The `project.properties` file contains a set of global settings, followed by the specific setting(s) for each type of component.

For each available component, there can be one or more configuration files. In `project.properties` file we need to enter the paths to these configurations files for each component. Additionally, in these files you can configure project specific settings. For example, you can configure **IBM AD Batch Server** to run only **Rule Based** reports for a specific project. Or you can configure **IBM AD Batch Server** to save the reports for one project in one folder and for other project in another folder. For more optional settings, see the project specific settings in the corresponding section for each component.

Global Settings

The global settings specify the projects on which the **IBM AD Batch Server** will operate and which components will run on the specified projects.

The `project.properties` file is where you can set the following global properties.

- Comma separated list of project names that will be the only ones considered for this service.

  Important: If no value is set for this parameter no report will be generated. ***** means all projects.

```
projects.whitelist=<Project>
```

- Comma separated list of project names that will be excluded for this service (lower priority than whitelist). Leave empty or remove property if not needed.

```
projects.blacklist=<Project>
```

- Comma separated list of component names that will be the only ones considered for this service. Ex.: **index** must be added as a component.

```
components=index,gdbImport,annUpdate
```

  Optional components can be considered for this service.

Table 7. Optional Components	
Component	**Description**
ruleBased	The **Rule Based** component generates reports for the resources specified in the configuration files according to the rules and parameters defined in the corresponding configuration files. **Note:** If the **Rule Based** component is used, make sure that the `ruleBased.properties` file is configured. For more information, go to “ruleBased.properties File” on page 87.
reports	The **Reports** component is used to generate the complexity reports. For more information, go to “Complexity Reports” on page 359.
cobolPP jclPP pl1PP	The **cobolPP**, **jclPP**, and **pl1PP** components generate the expanded sources for Cobol, JCL, and PL/I. For more information, go to “View Expanded Source” on page 318.
wsmetrics	The **wsmetrics** component is needed only if IBM ADDI Extension is used on the system. **Note:** Additionally, in order for the **wsmetrics** component to be executed, make sure that the **gdbImport** component is included in the components list of this service.

Important: It is vital that you set the parameters for the **Search Index**, **Annotations** and **GraphDB** components. Failing to do so will make some of the analysis impossible to generate in **IBM AD Analyze**.

Annotations

IBM AD Batch Server reads the configurations present under Annotations Database configurations from **IBM AD Configuration Server**. The configurations were set in `server.properties` in previous versions, and according to those settings it creates, if necessary, the repository where all user-defined annotations for all projects are stored.

Note: To make sure that Annotations Database was created in your database repository server instance check the following steps.

1. Go to **IBM AD Batch Server** installation folder and search for the `conf` folder.
2. Open the `server.properties` file that can be found in `conf` folder and verify that **IBM AD Configuration Server** host and environment are correctly filled in. For more information, see “Server Properties” on page 78.
3. Go to **IBM AD Configuration Server** > **Environment** > **Configurations** > **Annotations Database** and make sure that **Database name**, **Database schema**, and **Database server** settings are correctly filled in.
4. Start IBM AD Batch Server.
5. Go to the database repository instance set-up under Annotations Database configurations from **IBM AD Configuration Server** and you should find the database having the name mentioned in Database name field.

   Note: By default, **IBM AD Batch Server** will preserve in Annotations Database only those annotations for projects which are present in the projects list and will delete all other references/annotations. If you want to keep all annotations for all projects including the projects which are deleted at some point, uncomment and enter `false`.

```
## if true, all references of projects that do not appear in projects list will be deleted from annotations database.
(the default behavior)
```

```
#annUpdate.manageDeletedProjects=true
```

Application Settings

This setting allows the user to specify which components will run on a specific project.

Application Project Settings

This is the list of components that will execute for the project. **Empty** means no component will be executed; use ***** for all. Use a separate line for each project for which you want to define the components list.

```
#project.<projectName>.components=
```

Index Project Files

The **Index** component indexes the resources of a project so that a search in resources can be performed in **IBM AD Analyze** client, by using **Search in Files** analysis.

Important: Make sure that the Index location is a shared folder and has the appropriate permissions, for the situation when **IBM AD File Service** is installed and configured on a different machine than the Index location so that the **Search in Files** analysis can be used by all **Analyze Client** users. For more information, see the "Search in Files Analysis Tab" on page 316 section.

Index Global Settings

The Index component indexes the resources of a project so that a Search in resources can be performed in **IBM AD Analyze Client**, by using Search in Files analysis.

```
index.indexFolder=\\\\path\\<Folder>\\Index
```

Note:

- Starting with IBM AD V5.1.0.3, **IBM AD Search Service** is responsible with the access to the indexed data. Whether the authorization/authentication feature is used or not, the folder path in which the indexes are generated needs to be accessible both for **IBM AD Batch Server** and **IBM AD Search Service**. The path where the index data is generated needs to be added in the `conf.yaml` file of the **IBM AD Search Service**, where the *indexPath* parameter is present.
- For the accessible path defined in the `project.properties` file, backslashes must be doubled (****) and spaces in the path must have a single backslash as a prefix.

  Examples for the *index.indexFolder* parameter:

  - `index.indexFolder=\\\\path\\<Folder>\\Index`
  - `index.indexFolder=C:\\ibm\\Index`

Important:

- The folder path in which the indexes are generated needs to be accessible both for **IBM AD Batch Server** and **IBM AD Search Service**.
- The Index location is used during the configuration process of **IBM AD Search Service**.

The **`index.encoding`** parameter can be used to specify the encoding that is used to read target files. If this parameter is not configured in the `project.properties` file, the default encoding is UTF-8. You can set the value to another encoding according to IANA standards. This setting applies only to IBM AD Batch Server.

```
index.encoding=UTF-8
```



Warning: The encoding must match the one that is used in source files. For more information about how to configure the source file encoding, see the Chapter 5, "Configuring," on page 33 section.

Once the indexes are created in the default location, if there is a need to change this default folder, the following parameter should be used with value **true**.

Important: Starting with IBM AD V5.1.0.3, the `index.forceFolder` parameter is no longer present in the `project.properties` configuration file. The **IBM AD Search Service** is now responsible with the access of the indexed data when Search in files analysis from **IBM AD Analyze Client** is performed. If you want to change the index folder path you can modify the *index.indexFolder* parameter, but in the same time you need to make sure that the same path is used in the `conf.yaml` file of the **IBM AD Search Service**, where the *indexPath* parameter is present.

```
index.forceFolder=true
```

If a folder matches one of the patterns that are specified below, it will not be indexed.

```
index.exclude.folderPattern=pattern,(pattern)...
```

If a file matches one of the patterns that are specified below, it will not be indexed.

```
index.exclude.filePattern=pattern,(pattern)...
```

Index Project Settings

All the parameters in the global settings can also be configured individually for each project. The parameters set for each project override the global parameters.

The output folder where the index for this project is stored must be unique per project.

Note: This setting does not override database entry!

```
project.<projectName>.index.indexFolder=
```

This is how you can override the database folder and create the index in a new location:

Note: Starting with IBM® AD V5.1.0.3, the `project.<projectName>.index.forceFolder` parameter is no longer present in the `project.properties` configuration file. The **IBM AD Search Service** is now responsible with the access of the indexed data when Search in files analysis from**IBM AD Analyze Client** is performed. If you want to change the index folder path for each project, you can modify the *project.<projectName>.index.indexFolder* parameter, but in the same time you need to make sure that the same path is used in the `conf.yaml` file of the **IBM AD Search Service**, where the *projects* section is present.

```
project.<projectName>.index.forceFolder=(true|false) default false
```

This is how you can specify extra folders to be indexed (apart from project folder):

Note: Make sure that all backslash characters in all paths are doubled: ****.

```
project.<projectName>.index.extraSources=path, path
```

If a folder matches one of these patterns, it will not be indexed.

```
project.<projectName>.index.exclude.folderPattern=pattern(,pattern)...
```

If a file matches one of these patterns, it will not be indexed.

```
project.<projectName>.index.exclude.filePattern=pattern(,pattern)...
```

ADI Indexing

The **ADI Index** component will index the resources of a project so that the **Business Rules Discovery (BRD)** feature, from **IBM® ADDI Extension** application, can display code snippets.

1. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\conf folder, open the adi.properties file, and make sure that the file contains the following URL for the ADI indexing service.

```
##URL for the ADI indexing service
es.serverUrl=https://localhost:9444/addi/es
```

   Note: Please do not update the ADI indexing service URL, unless otherwise instructed by the Admin that sets-up and configures the IBM ADDI extension.

2. In the project.properties file, which is located in the \conf folder, enter the **adidx** component name.

```
## list of components that will be executed
## example components:
ruleBased,reports,index,gdbImport,annUpdate,cobolPP,jclPP,pl1PP,wsmetrics,adidx
## empty means none will execute!!, use * for all
components=adidx
```

Cobol, JCL and PL/I Preprocessing

View expanded source functionality in **IBM AD Analyze** can only be available if a preprocessing operation has been performed on the analyzed resources. Preprocessing of **Cobol** resources implies expanding all include files and applying all **REPLACE** compiler directives. For **JCL**, pre-processing implies expanding all **JCL** procedures and **JCL** include files.

Preprocessing of **PL/I** resources implies expanding all *include* files, having the following supported syntax:

```
File inclusion: %INCLUDE File, %XINCLUDE File, ++INCLUDE File, -INC File;
```

```
Library member inclusion: %INCLUDE MEMBER, %INCLUDE LIB (MEMBER).
```

The **Cobol**, **JCL** and **PL/I** preprocessing parameters allow the user to customize the details of the operation.

Cobol Preprocessor Global Settings

This parameter specifies the encoding used to read **Cobol** programs and copybooks when generating the preprocessed source. The default value is **UTF-8**, but any Java encoding is accepted. For example, for Japanese ideograms use **Shift_JIS**.

```
cobolPP.encoding=Shift_JIS
```

If the source code contains tabs instead of blank spaces, the preprocessor should know how many spaces to put instead of the tabs. The tab size default value is *4*, allowed values are *>= 1*.

```
#cobolPP.tabSize=4
```

Threads parameter is used to set the maximum number of threads per task to execute when preprocessing. The default value is *5*, allowed values are *>= 1*.

```
#cobolPP.threads=5
```

Emulate prefix parameter dictates whether or not to handle **PREFIX** and **-INC** directives when preprocessing, the default value is **true**.

```
#cobolPP.emulatePREFIX=true
```

Full path parameter dictates whether to specify in the inserted comment the full path of the expanded copy file in the preprocessed source, default value **true**. If set to **false**, only the name of the copy file and its content will be inserted in the preprocessed source.

```
#cobolPP.copy.fullPath=true
```

JCL Preprocessor Global Settings

This parameter specifies the encoding used to read files in order to generate preprocessed source. The default value is **UTF-8**, but any Java encoding is accepted. For example, for Japanese ideograms, use **Shift_JIS**.

```
jclPP.encoding=Shift_JIS
```

Threads parameter can be used to set the maximum number of threads per task to execute when preprocessing. The default value is *5*, but the allowed values are *>= 1*.

```
#jclPP.threads=5
```

PL/I Preprocessor Global Settings

This parameter specifies the encoding used to read files in order to generate preprocessed source. The default value is *UTF-8*, but any Java encoding is accepted. For example, in case of Japanese ideograms, use *Shift_JIS*.

```
pl1PP.encoding=Shift_JIS
```

Threads parameter can be used to set the maximum number of threads per task to execute when preprocessing. The default value is *5*, but the allowed values are *>= 1*.

```
pl1PP.threads=5
```

Important: If the source code contains characters or digits beyond the column 72 (in the inactive zone, for example *BG100 : PROCEDURE OPTIONS (MAIN);[ ]* ***00020002***) then the next additional setting must be made:

```
pl1PP.line_max_len=73
```

Rule Based Analysis Reports

The **Rule Based** component will generate reports for the resources specified in the configuration files according to the rules and parameters defined in the corresponding configuration files.

A configuration file is line based text organized as key/value pairs. General properties of a configuration file include:

- empty lines are ignored.
- lines that start with **#** are ignored (comments).
- keys do not contain spaces or equal sign **=**.
- a value starts at the first non-space character after **=**and finishes at the end of the line.
- values can expand on multiple lines by means of a single backslash character followed by newline.

Rule Based Global Settings

File specifying what information will be displayed on the first page of the generated reports.

```
ruleBased.configFile=ruleBasedConfig.properties
```

File containing the actual definition of the rules.

```
ruleBased.rulesFile=ruleBased.properties
```

Encoding of the file containing the actual definition of the rules, the default encoding is **UTF-8**, to modify, uncomment and enter desired encoding.

```
ruleBased.rulesFile.encoding=UTF-8
```

Option to generate an individual report for each source file where a rule failed, default value **true**.

```
ruleBased.reportsPerSource=true
```

Option to generate for each source file an additional file (named `controlTotal`) containing the number of rules that failed for that file. Valid only if the previous parameter, **`reportsPerSource`**, is set to **true**. The default value is **true**.

```
ruleBased.generateTotals=true
```

Output folder for rule based reports generated by source based rules. The default folder is the data folder, to set a different folder, uncomment and enter the path to that folder. The default format for the reports is `*.`PDF.

```
ruleBased.reportsFolder=
```

Output folder for reports generated in `*.`CSV format. The default folder is the data folder, to set a different one, uncomment and enter the path to that folder.

```
ruleBased.csvFolder=
```

Output folder for `controlTotals` files generated by source based rules. The `controlTotals` files are the files containing the number of rules that failed for a specific source file. The default folder is the data folder, to set a different folder, uncomment and enter the path to that folder. This parameter is only valid only if `generateTotals` is set to **true**.

```
ruleBased.totalsFolder=
```

Default suffix for the `controlTotals` file. To set a different suffix, uncomment and enter the desired suffix.

```
ruleBased.totalsSuffix=.CDGSTDS.TOTALS
```

Prefix for the `controlTotals` file, by default the `controlTotals` do not carry a prefix, to use a prefix uncomment and enter the desired prefix.

```
ruleBased.totalsPrefix=
```

Rule Based Project Settings

All the parameters in the global settings can also be configured individually for each project. The parameters set for each project override the global parameters.

- Absolute path to rule configuration file used for Rule Based analysis component.

  ```
  project.<projectName>.ruleBased.rulesFile=projectName.rules.properties
  ```

- Option to generate an individual report for each source file where a rule failed. The default value is **true**.

  ```
  project.<projectName>.ruleBased.reportsPerSource=
  ```

- Option to generate for each source file an additional file (named controlTotal) containing the number of rules that failed for that file. Valid only if **`reportsPerSource`** is set to **true**.

  ```
  project.<projectName>.ruleBased.generateTotals=
  ```

- Output folder for rule based reports generated by source based rules. The default folder is the data folder. To set a different folder, uncomment and enter the path to that folder. The default format is `*.PDF`:

```
project.<projectName>.ruleBased.reportsFolder=
```

- Output folder for `*.CSV` files generated by data based rules. The default folder is the data folder. To set a different folder, uncomment and enter the path to that folder:

```
project.<projectName>.ruleBased.csvFolder=
```

- Output folder for controlTotals files generated by source based rules. The default folder is the data folder. To set a different folder, uncomment and enter the path to that folder. Valid only if **`generateTotals`** is set to **true**.

```
project.<projectName>.ruleBased.totalsFolder=
```

ruleBasedConfig.properties File

All data set in this file will appear on the cover page of the reports.

- Name of the owner of the report.

```
owner=
```

- Name of the organization.

```
organization=
```

- Role of report owner within the organization.

```
organizationRole=
```

- Image that will appear on cover page of the report.

```
coverImage=
```

- Project name that will appear on cover page; if not set here, the name of the project given as input will be used.

```
projectName=
```

ruleBased.properties File

This file specifies the actual rules on which the **Rule Based** reports are generated, different parameters for the rules as well as the source files on which the **Rule Based** analysis will run.

Here are some specific syntax elements of the `RuleBased.properties` file:

- <number>: is a positive integer.
- <identifier>: alphanumeric text with no spaces.
- <path>: absolute path to a file on disk.
- <class>: qualified class name for a rule.
- <text>: any string of characters on one line. Line continuation (single backslash at the end of line) can be used for multiple lines. A syntax comment is introduced hereby. It is not part of the file format.
- {key}: the value of the key as found in the file.

Rules can be parametrized, inputs being one of the optional parameters which can be set. Input values can be a list of data from the database based on a query. Rule characteristics:

- **id** - unique number to identify an input; it is used to create all subsequent keys for current input.

- **key** - unique key used for substitution in query rules.
- **name** – input name.
- **query** - query to obtain a list of values from the repository which will act as parameters.
- **firstColumnType** - declare the type for the first column of the query. The default value is numeric. Accepted, string.

Parameter values can be simple values (numbers or text) and are also optional. Parameter characteristics:

- **id** - unique number to identify a parameter; it is used to create all subsequent keys for current parameter.
- **key** - unique key used for substitution in query rules.
- **name** – parameter name.
- **pattern** - if parameter is a simple value, pattern can be used to validate the data the user entered.
- **length** - if parameter is a simple text value, length can be used to verify the max length of the data.
- **value.lessThan** - for numeric simple values it can be used for validation.
- **value.greaterThan** - for numeric simple values it can be used for validation.
- **prefix** - text to prefix the user chosen data (transforms the parameter as text).
- **suffix** - text to be appended to the user chosen data (transforms the parameter as text).

Rules can be organized in groups according to specific criteria. Rule group characteristics:

- **id** - uniquely identifies a group; it is used to create all subsequent keys for current group.
- **name** - name to display for the group.
- **parent** - a {group.id} value of another group that contains this group.
- **description** - additional information on the group displayed in the application.

Rules are the basis of the Rule Based analysis action. Rule characteristics include:

- **id** -unique number that identifies a rule; it is used to create all keys for current rule.
- **name** – the name that is visible for the user. This will appear in the report.
- **sourceBased** - classifies the rule. The report will be generated in `*.`PDF format if **sourceBased** is set to **true**. If **sourceBased** is set to **false**, the report will be generated in `*.`CSV format. The default value is **true**.
- **weight** – a number used to calculate the weight of a certain rule, appears in the report summary. The default value is **1**.
- **query** and **selectiveQuery** - paths to query files.
- **class** - qualified class name for a rule implementation.
- **groups** - list of group ids that will show the rule.
- **description** - long text that describes the rule to the end-user.
- **params** - list of parameter keys that are required by this rule.
- **inputs** - list of input keys that are required by this rule.

Reports

The **Report** component will generate the reports specified in the configuration file.

Report: Global Settings

Important: The global settings used by these reports is set in the `project.properties` file.

Absolute path to the file containing a description of the report headers: `#reports.configFile=${install.dir}.reportsConfig.properties`.

Comma separated list of report codes that will be generated.

```
## 15103=Halstead
## 15104=Cyclomatic
## 15105=Maintainability
## 15106=Heuristic
## 15108=Unused Routines Within Programs
#reports.whitelist=
```

Comma separated list of report codes that will NOT be generated (lower priority than whitelist).

```
#reports.blacklist=
```

Output folder for generated reports. The default folder is the data folder, to change, uncomment and enter absolute path to desired folder.

```
#reports.outputFolder=
```

Option to generate a unique report for all resources. The default value is **true**. To generate a separate report for each resource, enter **false**.

```
#reports.coalesceInputs=true
```

Report: Project Settings

Important: The global settings used by these reports is set in the `project.properties` file.

All the parameters in the global settings can also be configured individually for each project. The parameters set for each project override the global parameters.

Comma separated list of report codes that will be generated for the project.

```
## 15103=Halstead
## 15104=Cyclomatic
## 15105=Maintainability
## 15106=Heuristic
## 15108=Unused Routines Within Programs
#project.<projectName>.reports.whitelist=
```

Comma separated list of report codes that will NOT be generated (lower priority than whitelist)

```
#project.<projectName>.reports.blacklist=
```

Absolute path to custom report configuration file for the project.

```
#project.<projectName>.reports.configFile==${install.dir}.projectName.reportsconfig.properties
```

Option to generate a unique report for all resources in the specified project. The default value is **true**. To generate a separate report for each resource, enter **false**.

```
#project.<projectName>.report.coalesceInputs=<true|false>
```

reportsConfig.properties File

This file allows the user to determine what information will appear on the first page of the reports, whether to show the appendix of a report, as well as where to find the configuration files which allow the customization of different report types.

The data set in this file will appear on the cover page of the reports.

- Name of the owner of the report.

  ```
  owner=
  ```

- Name of the organization.

```
organization=
```

- Role of report owner within the organization.

```
organizationRole=
```

- Image that will appear on cover page of the report.

```
coverImage=
```

- Project name that will appear on cover page; if not set here, the name of the project given as input it will be used.

```
projectName=
```

- Show appendix in reports; the default value is **false**.

```
showAppendix=
```

- Absolute path to the file containing custom settings for reports (values for weights, colors for pie charts, etc.)

```
halstead.settingsFile=
heuristic.settingsFile=
cyclomatic.settingsFile=
maintainability.settingsFile=
unusedRoutines.settingsFile=
```

cyclomatic.properties File

This file contains settings which allow the customization of the **Cyclomatic** reports.

Note: This configuration file is located in the `conf/reports` folder.

The colors used in the pie charts present in the report:

```
#simpleColorPreference=0,255,0
#mediumColorPreference=250,196,69
#complexColorPreference=221,0,0
```

Values for thresholds used to separate simple/medium/complex groups:

```
#simpleThresholdPreference=10
#mediumThresholdPreference=50
```

halstead.properties File

This file contains settings which allow the customization of the **Halstead** reports.

Note: This configuration file is located in the `conf/reports` folder.

By default, all statements are considered operators. If you want to set a specific statement as operand, enter the statement id (from the **Statement** table in the database) then enter **false**.

```
##stmt id = check state
#if not present, falls to default. In default file, default value = check
#29=false
#28=false
```

heuristic.properties File

This file contains settings which allow the customization of the **Heuristic** reports.

Note: This configuration file is located in the `conf/reports` folder.

Colors used in pie charts present in the report:

```
#simpleColorPreference=0,255,0
#mediumColorPreference=250,196,69
#complexColorPreference=221,0,0
```

Values for thresholds used to define simple/medium/complex groups:

```
#simpleThresholdPreference=200
#mediumThresholdPreference=500
```

By default, all statements are assigned a weight of **1**, if you want to assign a different weight for a specific statement, enter the statement id (from the **Statement** table in the database) and then desired weight value.

```
##stmt id = weight value
#if not present, falls to default. In default file, default value = 1
#1815=100
#1814=1000
#109=2
#108=3
```

maintainability.properties File

This file contains settings which allow the customization of the **Maintainability** reports.

Note: This configuration file is located in the `conf/reports` folder.

Colors used in pie charts present in the report:

```
#complexColorPreference=0,255,0
#simpleColorPreference=221,0,0
#mediumColorPreference=250,196,69
```

Values for thresholds used to define simple/medium/complex groups:

```
#simpleThresholdPreference=65
#mediumThresholdPreference=85
```

By default, all statements are considered operators. If you want to set a specific statement as operand, enter the statement id (from the **Statement** table in the database) then enter **false**.

```
#stmt id=false
#29=false
#28=false
```

unusedRoutines.properties File

This file contains settings which allow the customization of the **Unused Routines** reports.

Note: This configuration file is located in the `conf/reports` folder.

By default, **Exit Procedures** will NOT be shown in the report, to include the **Exit Procedures** in the generated report, uncomment and enter **false**.

```
#filterExitProcedures=true
```

Web Service Metrics Component

Web Service Metrics component generates input data for **IBM AD Web Service**. The metrics values are packaged into **JSON** objects.

IBM AD Web Service

IBM AD Web Service collects the data provided by **Web Service Metrics** component and prepares it for delivery.

IBM AD Web Service parameters are stored in a `webservice.properties` file available in **IBM AD Batch Server** installation folder.

The following parameters can be set:

1. The IP of the computer where **IBM AD Web Service** is installed. The default value is **0.0.0.0** (all interfaces). Another valid value is the external IP of the computer.

```
## web service host
#host=
```

2. The port no. to be used by the **IBM AD Web Service**; the default port is **8090**, if no other value is specified the default value will be used.

```
## web server port
## default: 8090
#port=
```

3. The administrative port number to be used by **IBM AD Web Service**; the default administrative port is **8099**, if no other value is specified the default value will be used.

```
## web service administrative port
## default: 8099
#admin-port=
```

4. Absolute path to the **Web service** directory.

```
## web service directory
web-dir=webdata
```

5. Absolute path to the **JSON** files directory.

```
## json files directory
rest-dir=wsmetrics
```

6. Absolute path to file containing the Web service authentication data.

```
## authentication info for the web service
auth-file=auth.properties
```

7. Absolute path to file containing the SSL certificate.

```
##ssl keystore file
Keystore-file=keystore.jkl
```

8. Absolute path to the file containing log configuration data.

```
## log configuration file
log-conf-file=webservice.log4j.properties
```

IDCAMS Preprocessor

To display the dataset usage via **IDCAMS** utility in **Dataset Usage** in **Jobs**, a preprocessing operation must be performed on the analyzed resources. Running **IDCAMS** preprocessing implies saving data from all available **JCL** jobs that execute **IDCAMS** utility into the database repository project repository. **IDCAMS** preprocessing does not need to be added as a separate component, it runs on each project added in the <project.whitelist>.

Global Settings

This parameter specifies the encoding used to read **JCL** jobs and **CTRL** files when parsing the sources for all available projects. The default value is **UTF-8**, but any Java encoding is accepted.

You can setup this encoding in the `project.properties` file.

For example, in case of Japanese ideograms use **Shift_JIS**:

```
ams.encoding=Shift_JIS
```

Project Settings

This parameter specifies the encoding used to read **JCL** jobs and **CTRL** files when parsing the sources for a specific project. The default value is **UTF-8**, but any Java encoding is accepted.

You can setup this encoding in the `project.properties` file, using the following entry:

```
project.<projectName>.ams.encoding=UTF-8
```

Importing Multiple Projects when IBM AD Batch Server is Installed and Configured on Linux/zLinux

Perform the following configurations if you want to import multiple projects when **IBM AD Batch Server** is installed and configured on Linux/zLinux.

Important: The following configurations need to be performed for versions up to **IBM AD V5.1.0.7** as follows:

By modifying the Global Settings

1. On the machine where IBM AD Batch Service is installed, edit `/etc/sysctl.conf` file and add the following line.

   ```
   fs.file-max = 65536
   ```

2. Apply the changes by using the following command.

   ```
   sudo sysctl -p /etc/sysctl.conf
   ```

Per user limit

1. Edit as root the following system configuration file.

   ```
   % sudo vi /etc/security/limits.conf
   ```

2. Modify the values for **AD_user**.

   Note: We assume that **JBOSS** was launched with the system user **AD_user**.

   ```
   AD_user           soft    nofile          4096
   AD_user           hard    nofile          8192
   ```

3. If you want to raise the limits for all users, modify the values as follows.

   ```
   *           soft    nofile          4096
   *           hard    nofile          8192
   ```

4. Open a new session with the user that starts **OrientDB** and check whether the changes are valid.

   ```
   % su AD_user (orient_user/batch_user)
   % ulimit -n
   1024
   ```

5. If the changes were not taken into account perform the following actions:

   a. Edit `/etc/pam.d/su:` as follows.

      ```
      % sudo vi /etc/pam.d/su
      ```

   b. Uncomment the following line.

      ```
      session    required   pam_limits.so
      ```

c. The changes are taken into account at the next login.

```
% su AD_user
% ulimit -n
4096
```

STEP 9. Configuring IBM AD Search Service

Follow the configuration steps that are needed to have up and running **IBM AD Search Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Search Service available in IBM AD Configuration Server

   Note: IBM AD Search Service is necessary for the Search in Files functionality to work. It is mandatory to have **IBM AD File Service** installed, configured and up and running.
3. Restart IBM AD Search Service

1. Configure the parameters that are present in the `conf.yaml` file

On the machine where **IBM AD Search Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

   ```
   #if communication should be secured with TLS
   https: false
   ```

   b. If the **https** parameter is set to *true*, a secured communication is used.

   Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

   ```
   #if communication should be secured with TLS
   https: true
   ```

   Note: If the **https** parameter is set to *true*, an additional step needs to be performed. Locate `startServer.bat` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/` and replace the following line:

   ```
   set tlsoptions=
   ```

   with:

   ```
   SET keystorepath=<"path_to_keystore">
   SET keystorepass=<"password_of_keystore">
   set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
   -Djavax.net.ssl.keyStorePassword="%keystorepass%"
   ```

   Where:

   - **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Search Service**.

- **Keystore password** is the keystore password.

2. Set the **authSrv** parameter as follows:

   a. If the value of the **https** parameter is set to *true*, add the URL of **Authentication Server (DEX)** where **authSrv** parameter is present. **Authentication Sever (DEX)** that belongs to the **IBM AD** package is used. For more information, see "STEP 4. Configuring Authentication Server (DEX)" on page 111. Example:

   ```
   #authentication server URL
   authSrv: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
   ```

   b. If the value of the **https** parameter is set to *false* and the **Authorization and Authentication** feature is *enabled*, add the URL of **Authentication Server (DEX)**. Example:

   ```
   #authentication server URL
   authSrv: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
   ```

3. Set the **disableAuth** parameter to *false*. The *false* value keeps enabled the authentication.

   ```
   #disable authentication/authorization. allow all files to be sent
   disableAuth: false
   ```

4. Optionally, the main path where the indexes are created for each project, can be overwritten by the following configuration. This path needs to be identical to the one present under `<IBM ADDI Installation Folder>/IBM Application Discovery Batch Server/conf/project.properties`, where the **project.projectName.index.indexFolder** parameter is present.

   ```
   #optional
   #overrides indexPath for a specific project.
   projects:
      - project: Project1
        path: C:\Index\Indexes\Project1
   ```

2. Make IBM AD Search Service available in IBM AD Configuration Server

IBM AD Search Service is responsible with the access to the indexed data. The folder path in which the indexes are generated needs to be accessible for both **IBM AD Batch Server** and **IBM AD Search Service**.

After **IBM AD Search Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Search Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Search Service**. The **Search Service** settings page is displayed.
2. Add the path where the **indexes** are created. This path needs to have as an endpoint the same folder where the **indexes** are stored. Example:

   ```
   \\server01\Indexes
   ```

3. Optionally, the main path where the indexes are created for each project, can be overwritten. Click **Add**, select the target project, and add the path where the indexes are created for the target project.
4. Click **Save**.

3. Restart IBM AD Search Service

- **On Windows**

  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Search Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.

  2. Click **Start**, select **Run**, type **`services.msc`** and start **IBM Application Discovery Search Service**.

3. If the service does not start, check the `search.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/log` folder.

- **On Linux**

1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/` and locate the `startServer.sh` file.
2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

```
chmod +x startServer.sh
```

3. If the service does not start, check the `search.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/log` folder.

STEP 10. Configuring IBM AD Cross Applications Service

IBM AD Cross Applications Service is an **additional** service that needs to be configured to show calls between different mainframe projects that have their databases on the same DB instance in **IBM AD Analyze Client**.

Follow the configuration steps that are needed to have up and running **IBM AD Cross Applications Service**:

1. Configure the parameters that are present in the `conf.yaml` file
2. Make IBM AD Cross Applications Service available in IBM AD Configuration Server
3. Restart IBM AD Cross Applications Service

1. Configure the parameters that are present in the `conf.yaml` file

On the machine where **IBM AD Cross Applications Service** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/conf/` and make sure that the `conf.yaml` is present. If the `conf.yaml` file is not present in the `/conf` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/sample-conf/` and copy the `conf.yaml` file in the `/conf` folder. Open the `conf.yaml` file by using a text editor and enter the desired values for the parameters that are detailed below.

Note: The parameters are represented in YAML as mappings that consist of a parameter key and the value that is associated to that key. The format of the mapping is the parameter key represented by a string, which is terminated by a trailing colon that is followed by a space. The value for that parameter key is represented by a string that follows the key's colon and space. Example:

```
my_parameter: my_value
```

1. Add the port of **IBM AD Configuration Server**.

```
## Coordination and Configuration Server port
## default 2181
ccs.server.port: 2181
```

2. Set the **https** parameter as follows:

   a. If the **https** parameter is set to *false*, a non-secured communication is used.

```
#if communication should be secured with TLS
https: false
```

   b. If the **https** parameter is set to *true*, a secured communication is used.

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
#if communication should be secured with TLS
https: true
```

Note: If the **https** parameter is set to *true*, an additional step needs to be performed.
Locate `startServer.bat` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/` and replace the following line:

```
set tlsoptions=
```

with:

```
SET keystorepath=<"path_to_keystore">
SET keystorepass=<"password_of_keystore">
set tlsoptions=-Djavax.net.ssl.keyStore="%keystorepath%"
-Djavax.net.ssl.keyStorePassword="%keystorepass%"
```

Where:

- **Path to keystore** is the path to the keystore that holds the certificate for **IBM AD Cross Applications Service**.
- **Keystore password** is the keystore password.

3. Set the **authSrv** parameter as follows:

   a. If the value of the **https** parameter is set to *true*, add the URL of **Authentication Server (DEX)** where **authSrv** parameter is present. **Authentication Sever (DEX)** that belongs to the **IBM AD** package is used. For more information, see "STEP 4. Configuring Authentication Server (DEX)" on page 111. Example:

   ```
   #authentication server URL
   authSrv: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
   ```

   b. If the value of the **https** parameter is set to *false* and the **Authorization and Authentication** feature is *enabled*, add the URL of **Authentication Server (DEX)**. Example:

   ```
   #authentication server URL
   authSrv: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
   ```

4. Set the **disableAuth** parameter to *false*. The *false* value keeps enabled the authentication.

```
#disable authentication/authorization. allow all files to be sent
disableAuth: false
```

5. Optionally, you can decide what project resolutions to display in the analysis, when more projects contain the same program definition.

```
#optional: force resolution target
#program name pattern from source project will be solved only in target project
restrictions:
 - pattern:  "Program1"
   from.project: Project1
   to.project: Project2
```

Where:

- **pattern** - represents the name of the program which has the source present in other projects.
- **from.project** - represents the project that contains the program without the source.
- **to.project** - represents the project resolution that you decide to show in the analysis.

2. Make IBM AD Cross Applications Service available in IBM AD Configuration Server

By adding a restriction in the **Cross Applications Service** page, you can decide what project resolutions to display in the analysis, when more projects contain the same program definition.

After **IBM AD Cross Applications Service** is up and running, go to **IBM AD Configuration Server** and make **IBM AD Cross Applications Service** available for the other IBM AD components as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Services** > **Cross Applications Service**. The **Cross Applications Service** settings page is displayed.
2. To add a new restriction, click the **Add** button. As a result, the **Add Restriction** page is displayed.
3. In the **Pattern** field, enter the name of the program that has the source present in other projects.
4. In the **From Project** field, choose the project that contains the program without the source
5. In the **To Project** field, choose the project resolution that you decide to show in the analysis.
6. Click **Save**.

3. Restart IBM AD Cross Applications Service

- **On Windows**
  1. Once the configuration is done, go to the **Dashboard** tab, in **IBM® AD Configuration Server**, click the menu button of **Cross Applications Service**, and select **Restart Service**.

     Note: Wait until the service is restarted, this can take a few minutes to complete.
  2. If the service does not start, check the `cross.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/log` folder.
- **On Linux**
  1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/` and locate the `startServer.sh` file.
  2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable:

     ```
     chmod +x startServer.sh
     ```
  3. If the service does not start, check the `cross.log` file under `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/log` folder.

Important: At this installation and configuration point, everything is put in place for having analysis available in **IBM AD Analyze Client**. To take full advantage of the available analysis functionality, you need to install **IBM AD Analyze Client**. For more information see, "Installing IBM AD Analyze Client" on page 27.

STEP 11. (Optional) Configuring IBM AD Analyze Server

This step is optional and not necessary unless Java source code will be included in one or more projects in this environment.

On Linux only, Open the `servermount.properties` file, which is located under `<installation folder>\IBM Application Discovery Batch Server/conf` folder and specify how the windows shared folders are mounted on the local files system, by using the following pattern:

```
\\\\machine IP\\WindowsSharedFolder=/home/user/LinuxFolder
```

Example:

```
\\\\192.168.56.57\\ProjectsSharedPathWindows=/home/user/ ProjectsSharedPathLinux
```

It is mandatory to mount the Remote Path from **Analyze Server Manager** > **Server Settings**.

Following are the configuration steps that are needed after **IBM AD Analyze Server** was installed.

- **Under Windows**: to access the configuration parameters, select **Start** > **Programs** > **IBM Application Discovery Analyze Server** > **IBM Application Discovery Analyze Server Manager**.
- **Under Linux**: to access the configuration parameters, go to `<Installation Path>\IBM Application Discovery Servers\IBM Application Discovery Analyze Server` and run `manager.sh`.

In the **Server settings** tab, the **Server properties** and **Server arguments** sections display default data that was entered when **IBM AD Analyze Server** was installed.

To configure **IBM AD Analyze Server**, follow the instructions below.

1. In **IBM AD Configuration Server**, go to **Configurations** > **Default** > **Analyze Servers** and fill in the following details:
   - **`Host`**: IP / Hostname where **IBM AD Analyze Server** is installed.
   - **`RMI Registry Port`**: 1099 (Default).
2. Configuring the server database – the Database settings tab:

   Select the **Database Settings** tab. In the Location area, fill in the following parameters:

   - **Server type** field is completed by default with *SQL server*.
   - **Server IP**: Enter the IP of the computer where SQL Server is installed.
   - **Server port**: Server port is the access port, by default, 1433 port is used.

   Important: Make sure that the IP address and the port number you set here are the same as the ones entered in the **Relational database servers** page of **IBM AD Configuration Server**.

   Database instance, this field must be used in case the default database name was not chosen at SQL Server installation time. In the Authorization area, fill in the following parameters:

   - **Database name**: Enter a name for the database.
   - **User and password**: Give a user and password that can be used to create the database.

   After you completed the details of the database, click **Create database** to create the database with the selected parameters. If the database was configured correctly, after **Test database** is clicked, a message with the installed DB version will be displayed. Click **Save** to apply the settings.

   You can also select an existing database. If the selected database belongs to an older version of **IBM AD** and the database structure is now obsolete, a message is displayed indicating the current version of the selected database. The user is given the option to upgrade the existing database. Press **Upgrade** if applicable. After the upgrade process was performed, press again **Test database** to make sure that the upgraded database is functional. The version of the upgraded database is presented and **Upgrade** button is no longer available. For incomplete or corrupted databases one of the following messages may be displayed: **Database x is not a valid database** or **Cannot extract relevant data from the database. Database may be nonexistent, obsolete or invalid**.

   After modifying the settings in any of the tabs, do not forget to press **Save** to apply them. An asterisk at the beginning of the title of a tab indicates that parameters in that tab were modified but not saved.
3. Specifying allowed **IBM AD Analyze Clients**:

   Note: This configuration applies only to Java projects.

   - **IBM AD Analyze** clients can be of two types: manager and user.
   - **IBM AD Analyze** client of the **manager** type, can create shared projects, build shared projects, and delete shared projects.
   - **IBM AD Analyze** client of the “user” type, can only import the shared projects and perform analysis.

Manager and user types are server-related attributes, which means that a server determines the type for a client connecting to that server by looking up the client IP in the configuration files. This means that a client can be a manager on one server and a user on another server.

a) To add a Manager to the Managers list: Click **Add** in the upper right corner of the **Access Settings** tab: **New Access Data** dialog window is displayed. Enter the IP of the computer of the user who will access the server as a Manager (the type of owner is selected by default) then press **OK** to add the new manager to the list of Managers. To delete one of the Managers from the list, select it then press **Delete**. If you want to allow access to all the projects on the server to all potential users, do not add any users to the List of Users. If you want to limit the access to the projects to a number of specific users, select **restrict user IP** then add all of them to the List of Users. Only users in the List of Users and List of Managers will have access to the projects shared on the server.

b) To add a user to the List of Users, click **Add** from the List of Users area of the **Access Settings** tab: the New Access Data dialog window is displayed. Enter the IP of the computer of the user who will access the projects as a User (the type of owner is selected by default) then press OK to add the new user to the list of Users.

Under Windows: start **IBM AD Analyze Server**: Click **Start** and then select **All Programs** > **IBM Application Discovery Analyze Server** > **Start IBM Application Discovery Analyze Server**.

Alternatively, to start the server: From the Start menu, choose **Programs** > **IBM Application Discovery Analyze Server** > **Start IBM Application Discovery Analyze Server Monitor** then go to monitor icon from the taskbar, right-click on the icon, and select **Start service**. When the server is running, the green arrow from **Server Monitor** icon indicates that the server is started.

Under Linux: Go to `<Installation Path>\IBM Application Discovery Servers\IBM Application Discovery Analyze Server` and run `StartServer.sh`. Please make sure that this process remains alive.

Important: For monitoring the **IBM AD Analyze Server** tasks, see "Finding Log Files" on page 537.

Configuring IBM AD Analyze Client

To configure **IBM AD Analyze Client**, follow the instructions below.

Eclipse startup is controlled by the options in `$ECLIPSE_HOME/eclipse.ini`. If `$ECLIPSE_HOME` is not defined, the default `eclipse.ini` in your Eclipse installation directory is used.

1. **OS-Dependent Configuration**: In case **Analyze Client** is installed on Windows Server (any version) or Windows 8/10, you need to edit the `eclipse.ini` configuration file and add the following line, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section.

```
-Dorg.osgi.framework.os.name=win32
```

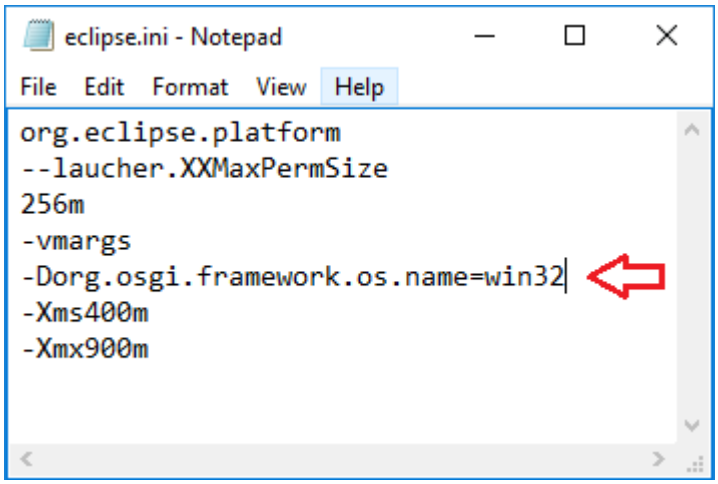


Figure 4. OS-Dependent Configuration

2. **Java-Dependent Configuration**: If an AD-supported IBM Java version is used as the system Java and you want to enable the TLS V1.2 connection, make sure to add the following lines in the `eclipse.ini` configuration file, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section.

```
-Dcom.ibm.jsse2.overrideDefaultTLS=true
-Djsse.enableCBCProtection=false
```

3. **TLS-Dependent Configuration**: If **IBM AD Analyze Client** is configured with **TLS** support, to have a secured communication between **IBM AD Analyze Client** and **IBM AD ZooKeeper**, make sure to add the following lines in the `eclipse.ini` configuration file, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section.

```
-Dzookeeper.client.secure=true
-Dzookeeper.ssl.keyStore.location=</path/to/keystore.jks>
-Dzookeeper.ssl.keyStore.password=password
-Dzookeeper.ssl.trustStore.location=</path/to/trustore.jks>
-Dzookeeper.ssl.trustStore.password=password
```

   Make sure that the `keystore.jks` and `trustore.jks` files are physically present on the machine where **IBM AD Analyze Client** is installed and configured. You might use the `keystore.jks` and `trustore.jks` files that were generated for **IBM AD ZooKeeper** if both **IBM AD Analyze Client** and **IBM AD ZooKeeper** are installed on the same machine. For more information, see "Generate a self-signed key certificate for IBM AD ZooKeeper" on page 203.

4. **OrientDB SSL-Dependent Configuration**: If **OrientDB** is configured with **SSL** in the `orientdb-server-config.xml` file, make sure to add the following lines in the `eclipse.ini` configuration file, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section.

```
-Dclient.ssl.enabled=true
-Djavax.net.ssl.keyStore=</path/to/client-keystore.jks>
-Djavax.net.ssl.keyStorePassword=password
-Djavax.net.ssl.trustStore=</path/to/client-keystore.jks>
-Djavax.net.ssl.trustStorePassword=password
```

   Important:

   Make sure that the `client-keystore.ks` file is physically present on the machine where **IBM AD Analyze Client** is installed and configured, and that the correct path to `client-keystore.ks` is added in the `eclipse.ini` configuration file. The file can be found in the server machine where IBM ADDI was installed, and then obtained from the path `<IBM ADDI Installation Folder>\security\<environment-id>\keystore.jks`.

   Make sure that the `orientdb.ks` file is physically present on the machine where **IBM AD Analyze Client** is installed and configured, and that the correct path to `orientdb.ks` is added in the `eclipse.ini` configuration file.

5. **IDz-Dependent Configuration**: If **IBM AD Analyze Client** is installed together with **IDz**, if you encounter any issue after the Graphs are generated (for example the screen freezes), make sure to add the following lines in the `eclipse.ini` configuration file, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section.

```
-Dsun.awt.noerasebackground=true
-Dsun.java2d.noddraw=true
```

6. **Cross Applications Service-Dependent Configuration**: If **IBM AD Cross Applications Service** is configured and up and running, make sure to add the following line in the `eclipse.ini` configuration file, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section.

```
-Dad.http.read.timeout=100
```

7. **Memory Management Configuration**: Eclipse must be configured to allow for optimized memory consumption. To configure Eclipse, edit the `eclipse.ini` file under the Eclipse installation folder and set the minimum memory parameter (marked –Xms), the maximum memory parameter (marked –Xmx). Following is an example of an `eclipse.ini` file containing parameters for optimized memory consumption.

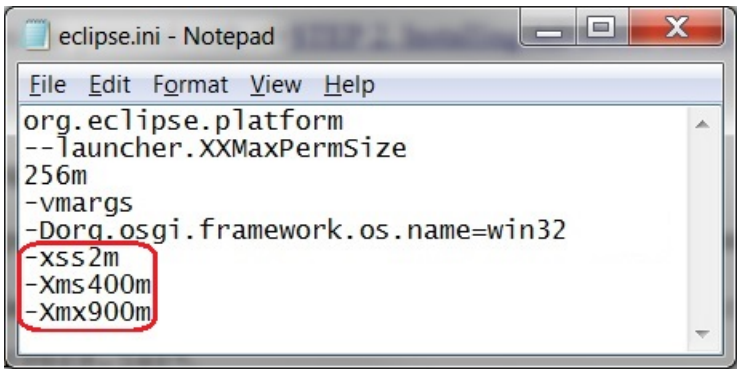


Figure 5. Memory Management Configuration

8. To use a specific language in the Eclipse interface, add the following parameter before the **-startup** parameter in the `eclipse.ini` file:

```
-nl
language
```

The **-nl** parameter has the following *language* values:

***Language* value**	**Language**
de	German
es	Spanish
fr	French
it	Italian
ja	Japanese
ko	Korean
pt_BR	Brazilian Portuguese
zh	Simplified Chinese
zh_HK	Traditional Chinese, Hong Kong
zh_TW	Traditional Chinese, Taiwan

9. Go to **IBM AD Analyze Client** main window and select **Window** > **Preferences** > **Application Discovery** > **Environment settings**.

   The following Environment identification settings are available:

   - **Host**, enter the hostname or the IP address of the computer where **IBM AD Configuration Server** is installed.
   - **Port**, enter the communications port number for **IBM AD Configuration Server**. If you are using the default port, enter 2181.
   - **Unique id**, enter the unique ID assigned by **IBM AD Configuration Server** to the environment you want to work with.
   - **Name**, enter the name of the environment with which you want to work, as defined in **IBM AD Configuration Server**.

10. Click **OK** and restart **IBM AD Analyze Client** (**File Menu** > **Restart**).
11. After restarting, a pop-up message displays the configurations that were made in **IBM AD Configuration Server**, and a restart is needed in order to be taken into account by **IBM AD Analyze Client**.

    Important: For monitoring the **IBM AD Analyze Client** tasks, see "Finding Log Files" on page 537.
12. In order to see the **Mainframe Analysis** projects, go to "Explore Projects Tab" on page 310.

Notice: At this point, all IBM AD components are up and running and ready for **Analysis**.

Configuring IBM AD Connect for Mainframe

Creating a z/OS Connection

Once IBM AD Connect for Mainframe and IBM AD Build Configuration are installed, open IBM AD Build Configuration, switch to the **zOS** tab, and create a new **zOS connection**.

In the **zOS Connection Wizard** dialog window: type the **Host IP** or **Host Name** and the port number as set in the listener PROC. Click **Save** and then perform **Test Connection**. If the test connection succeeds, click **Exit** to exit the wizard.

For more information, see "Configuring the z/OS Connection" on page 216.

Technical Configuration Process

Configuring the Listener PROC

The basic listener sample:

```
//IAYLSTNR PROC TCPIP=TCPIP,PORT=46000,MAXTASK=20,PROGRAM=IAYCONN,
//HOSTCP=00037,CLNTCP=00850
//LISTEN EXEC PGM=IAYLISPR,
//PARM='&TCPIP,&PORT,&MAXTASK,&PROGRAM,N,&HOSTCP,&CLNTCP,N',
//REGION=0M,TIME=NOLIMIT
//STEPLIB DD DSN=IAYV143.MVS110.LOADLIB,DISP=SHR <-- the agents loadlib
//CPEOUT DD SYSOUT=X,HOLD=YES
//IAYOUT DD SYSOUT=X,HOLD=YES
//IAYERR DD SYSOUT=X,HOLD=YES
//CPEOUTRS DD SYSOUT=X,HOLD=YES
//CPEOUTRQ DD SYSOUT=X,HOLD=YES
//CPESNAP DD SYSOUT=X,HOLD=YES
//SYSPRINT DD SYSOUT=X,HOLD=YES
//SYSUDUMP DD DUMMY
// PEND
```

Edit the listen PROC and modify the following in the STEPLIB card:

1. Change the default address space name for TCPIP if it differs at your site.
2. Decide on a communication port number for the listener by editing the PORT parameter in the PROC card. Note that an equivalent setup might also be done by using the IBM AD Build Configuration that is installed on the Windows system to correspond to the same the port. The port number might be unique and not used by other software. Any number can be selected.
3. Define the maximum number of tasks that can run in parallel (number of parallel requests from the PC server). Usually 5 - 10 might be enough.
4. Make sure that **REGION** and **TIME** parameters are set as in the listener sample.

   Note:

   - The **REGION=0M** parameter grants the AD Connect for Mainframe started task enough user storage for certain operations that require it.
   - The **TIME=NOLIMIT** parameter ensures that the AD Connect for Mainframe started task does not abend on a system code **S522** after exceeding the maximum wait time during periods of no client activity.
5. Accordingly, if IBM AD Connect for Mainframe's load library name would be changed as requested by your site standards, modify the STEPLIB card.

6. Define a new RACF user ID to assign to the AD Connect for Mainframe started task. The RACF user ID **must have an OMVS segment** to allow the started task to use TCP/IP services on z/OS. For more background on this requirement, refer to the following links:
   - The OMVS segment in user profiles
   - Requirement for an OMVS segment

   The new user ID must have appropriate access to all datasets referenced in the started task procedure JCL. After the user ID is defined, associate it with the started task procedure by using the following two RACF commands:

```
RDEFINE STARTED ADPROC.* STDATA(USER(AD_STC_USERID))
SETROPTS RACLIST(STARTED) REFRESH
```

   Note:
   - The **ADPROC** parameter is the member name that is assigned to the AD started task procedure JCL.
   - The **AD_STC_USERID** parameter is the new RACF user ID defined for its use.
7. For each IBM subsystem or SCM tool that is listed below, make the suggested change if you intend to use IBM AD Connect for Mainframe to retrieve data from this source.
   a. For **Db2**, add the following to the STEPLIB card:

```
// DD DSN=DSN910.SDSNLOAD,DISP=SHR <-- it should be changed if local load lib name is different at your site.
```

   b. For **IMS**, add the following to the STEPLIB card:

```
// DD DSN=IMS910.SDFSRESL,DISP=SHR <-- it should be changed if local load lib name is different at your site.
```

   c. For **CICS CSD**, add the following to the STEPLIB card:

```
// DD DSN=DFH320.CICS.SDFHLOAD,DISP=SHR <-- it should be changed if local load lib name is different at your site.
```

   d. For **MQ**, add the following to the STEPLIB card:

```
// DD DSN=CSQ700.SCSQAUTH,DISP=SHR <-- it shoud be changed if local load lib name is different at your site.
```

```
// DD DSN=CSQ700.SCSQLOAD,DISP=SHR <-- it shoud be changed if local load lib name is different at your site.
```

   e. For **CA Endevor**, add the following to the STEPLIB card, where **hlq** is the high-level qualifier for Endevor:

```
// DD DSN=hlq.ENDEVOR.AUTHLIB,DISP=SHR
```

   In addition, add the following DD cards (two cards per task, up to MAXTASK specified in the JOB card, with a different temporary file name per DD card). The DD cards might be numbered from 01 to **nn** to match the number of tasks defined in MAXTASK, modifying the numbers given below in red.

```
//* <-- FOR ENDEVOR - one DD for each task (from the MAXTASK parameter)
//* the number is running from 001 to nnn which is the MAXTASK number
//APIEX001 DD DSN=&&IAYIN01,SPACE=(3040,(40,40),,,ROUND),
// DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
// DCB=(BLKSIZE=20484,LRECL=2048,RECFM=VB)
//* <-- FOR ENDEVOR - one DD for each task (from the MAXTASK parameter)
//APIMS001 DD DSN=&&IAYOUT01,SPACE=(3040,(40,40),,,ROUND),
// DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
// DCB=(BLKSIZE=13300,LRECL=133,RECFM=FB)
```

   f. For **CA7**, add the following to the STEPLIB card:

```
// DD DSN=CA7.loadlib,DISP=SHR <-- it should be changed to site CA7 load lib name
```

8. When CA7 configuration is set to be used in batch mode – via dataset (so not in memory), add the following DD cards (+ card per task, up to MAXTASK specified in the JOB card):

```
//IAYINTR DD SYSOUT=(*,INTRDR)
//* Create as many IAYINTnn according to the number of MAXTASK parameter
//IAYINTnn DD SYSOUT=(*,INTRDR)
```

9. To connect to a local IBM Workload Scheduler (IWS, previously known as TWS) controller address space that is running on the same LPAR using a local subsystem interface, add the following DD Cards:

   a. `//EQQMLIB DD DISP=SHR,DSN=TWS851.SEQQMSG0` – it might be changed if the TWS Message Library has a different name on your site. If an user library is defined for TWS messages, it might come first and the two libraries names might be concatenated.

   b. `//EQQDUMP DD DISP=SHR,DSN=IAY.USER.TWS.EQQDUMP`. Create a dataset called `IAY.USER.TWS.EQQDUMP` with record format VBA, record length 84 and block size 3120. Allocation might be just few tracks.

   c. `//EQQMLGnn DD DISP=SHR,DSN=IAY.USER.TWS.MLOG.TASKnn`. Create datasets that are called `IAY.USER.TWS.MLOG.TASKnn` as sequential files with record format VBA, record length 125 and block size 1632, where **nn** goes from 01 to the number of tasks specified in MAXTASK.

   To connect to a remote IWS controller address space that is running on another LPAR in your sysplex or in a remote z/OS instance, see "Enabling TCP/IP Connection to a Remote IWS Controller Address Space" on page 199.

10. For TWS, add the following to the STEPLIB card:

```
// DD DSN=TWSnnn.SEQQLMD0,DISP=SHR <-- it should be changed if local load lib name is different at your site
```

11. For **ChangeMan ZMF**, add the following DD cards (two cards per task, up to MAXTASK specified in the JOB card. The files might be preallocated before the started task starts):

```
//* Create as many XMLINnn according to the number of MAXTASK parameter
//XMLIN01 DD DISP=SHR,DSN=IAY.CHGMAN.XMLIN01 <--FOR CHANGEMAN
//* Create as many XMLOUTnn according to the number of MAXTASK parameter
//XMLOUT01 DD DISP=SHR,DSN=IAY.CHGMAN.XMLOUT01 <--FOR CHANGEMAN
//IAYINTR DD SYSOUT=(*,INTRDR)
//* Create as many IAYINTnn according to the number of MAXTASK parameter
//IAYINTnn DD SYSOUT=(*,INTRDR)
```

12. For **Adabas**, add the following DD card (+ card per task up to MAXTASK specified in the JOB card. The files might be preallocated before the started task starts):

```
//IAYDR0nn DD DISP=SHR,DSN=IAYV142.NATURAL.DB.TASKnn
//* Create as many EZDR0nn according to the number of MAXTASK parameter. DSN can be any name
```

13. For **Natural**, add the following DD cards (+ card per task, up to MAXTASK specified in the JOB card. The files might be preallocated before the started task starts):

```
//IAYINTR DD SYSOUT=(*,INTRDR)
//* Create as many IAYINTnn according to the number of MAXTASK parameter
//IAYINTnn DD SYSOUT=(*,INTRDR)
//IAYCA0nn DD DISP=SHR,DSN=IAYV142.NATURAL.LB.TASKnn
//* Create as many IAYCA0nn according to the number of MAXTASK parameter. DSN can be any name
//IAYCB0nn DD DISP=SHR,DSN=IAYV142.NATURAL.MM.TASKnn
//* Create as many IAYCB0nn according to the number of MAXTASK parameter. DSN can be any name
```

Configuring IBM AD Connect for Mainframe Code Page

IBM AD Connect for Mainframe uses z/OS Unicode services to convert character data from one code page to another. There are two settings that tell the agent which code page to use for the host and the client:

1. HOST CODE PAGE
2. CLIENT CODE PAGE

In the JCL sample for IBM AD Connect for Mainframe, explanations on how to set the host and the client code are provided. The JCL sample is located in the **IAYLSTNR** member of the **SMP/E *.SIAYSAMP** library.

Note: The default value for the host is 00037 and for the client is 00850.

```
//* Note that starting with IBM AD 5.0.5, this proc must
specify                                    *
//* both the host CCSID, HOSTCP, used for encoding text on
the                                       *
//* mainframe, and client CCSID, CLNTCP, which is the
encoding                                        *
//* expected by IBM AD running on the workstation
```

```
PC.                                                    *
//* You should modify the default values if they are not
suitable.                                  *
//* Default HOSTCP=00037, and default
CLNTCP=00850.                                                  *
//*
    *
//* A list of valid CCSID values and descriptions is found
at:                                        *
//*
    *
//* https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.idad400/
ccsids.htm *
//*
    *
```

Japanese Language Host Code Pages

When host code page CCSID 00930 is used, it is important to note that the dollar sign *'$'* is converted to a backslash '\'. On Japanese Windows, the backslash is the same as the yen sign and it is used as the path delimiter. Windows does not allow creation of a file name when the yen sign is used.

If a mainframe member has the character *'$'* in its name, the member cannot be downloaded to the AD server on Windows by using the IBM AD Connect for Mainframe.

To prevent *'$'* from being changed to '\', the host code page CCSID 00939 can be used instead. However, other problems occur. The lowercase alphabet and single-byte Katakana are converted differently in host code pages CCSID 00930 and 00939.

If you require lowercase alphabet and single-byte Katakana in your analysis, it is advised to use host code page CCSID 00930 and to download the members that have in their name the character *'$'* manually, instead of using the IBM AD Connect for Mainframe.

Members that have in their name the character *'$'* need to be placed in a folder different from the folder to be synchronized by using the IBM AD Connect for Mainframe. The members need to be manually added to their respective Build Client virtual folders.

Configuration for CA-7

IBM AD Connect for Mainframe is configured to access CA7 by using the **CA7 CCI P-P** (program to program) interface. However, for extremely large batch schedules, it can consume large amounts of memory on loading. For these cases, it is preferable to use the AD-CA7 batch interface. In addition, if you want to perform an incremental update to retrieve only the CA7 data that has changed, since the initial bulk load (or last incremental update), **it is necessary to use the AD-CA7 batch interface**.

If you configured your AD installation to use the CA7 batch interface in earlier versions of IBM AD, be advised that beginning with IBM AD V5.1.0, the configuration process has changed and you need to review the remainder of this section. This change is necessary to allow for incremental updates from CA7 and to fully support this feature on systems running either JES2 or JES3.

The IBM AD Connect for Mainframe started task implements the CA7 batch interface by submitting a customized batch job to the internal reader. The batch job runs the necessary CA7 utilities and writes the data to a dataset. When complete, the final job step posts the AD Connect for Mainframe started task to indicate that the data is now available.

For this process to work, you must customize two sample jobs, provided in the **SIAYSAMP** dataset of the SMP/E distribution for IBM AD Connect for Mainframe. These jobs are saved in a suitable JCL library that can be read by the IBM AD Connect for Mainframe started task.

JCL Customization Process

All required samples are contained in the **SMP/E *.SIAYSAMP** library, as part of the SMP/E distribution of FMID HALT510, which includes two sample jobs and three Restructured Extended Executor (REXX) sample programs.

The three REXX programs, **IAYRXCA7**, **IAYRXCAC** and **IAYDSNDL**, don't need to be modified in any way. They can remain in the **SIAYSAMP** library or can be copied to another suitable library. All **SYSEXEC DD** names in the JCL sample need to be set to reference the location of these REXX programs. These changes are outlined in the following JCL customization steps.

Important:

- Make sure to tailor the JCL according to your site standards. Be advised that the **SYSIN** data might change according to your site standards.
- Make sure that this job is running on the **same** LPAR as the AD Connect for Mainframe started task. If your shop is configured in such a way that the job might be scheduled on another system, make sure that you use the **SYSTEM=*** parameter in the job card, as shown in the example. The ***** indicates the system that submitted the job. It does not represent a wildcard substitution for any system.
- Comments can be added or deleted to either of the sample jobs, but the size of either job must not exceed 80 lines (records).
- Do not add a job terminator card `'//'` to the end of any of the customized jobs.
- Do not turn on **ISPF PACK** for any customized job.
- Remove the `'<===Change(n)'` markers when done.

The following two samples jobs need to be configured for your site:

1. The **IAYCA7X** job

   When configured, this job is submitted by the AD Connect for Mainframe started task whenever a bulk load of all CA7 data is requested by the IBM AD Build Client or IBM AD Analyze server.

2. The **IAYCA712** job

   When configured, this job is submitted by the AD Connect for Mainframe started task whenever an incremental update of CA7 data is requested by the IBM AD Build Client or IBM AD Analyze server. The incremental update extracts only the changes that are made since the last incremental update, or the initial bulk load.

Specific configurations steps for the **IAYCA7X** job:

Note: Each step that is listed below is keyed by a change number (n). The sample JCL contains a change marker of the form `<===Change(n)` and indicates the location of the corresponding change.

1. Modify the job card to meet your site/system requirements.
2. Set **JOBLIB** to the IBM AD Connect for Mainframe **SMP/E *.SIAYAUTH** library.
3. Set the symbolic **SYSINDSN** parameter to a PDS/E library where this job can write a temporary CA7 report control card. Any suitable library can be used, but the member name **must** be set to **CA7T1%TK** as shown in the sample. Ensure that the IBM AD Connect for Mainframe started task has *UPDATE* access to this library.
4. Set the symbolic **CA7PRINT** parameter to a sequential dataset name. Any suitable dataset name can be user, but the final **%TK** prefix **must** be retained. The AD Connect for Mainframe started task replaces all instances of **%TK** with a unique task number (up to MAXTASK) so that each subtask can run in parallel with a unique dataset name. Ensure that the IBM AD Connect for Mainframe started task has *UPDATE* access to this sequential dataset.
5. **SYSEXEC DD** needs to point to a library, which contains the IBM AD REXX samples. You can use the IBM AD Connect for Mainframe **SMP/E *.SIAYSAMP** library if you want, since no user modifications need to be made to the REXX samples.
6. **SYSEXEC DD** needs to point to a library, which contains the IBM AD REXX samples. You can use the IBM AD Connect for Mainframe **SMP/E *.SIAYSAMP** library if you want, since no user modifications need to be made to the REXX samples.
7. **STEPLIB** needs to point to your **CA7 CAL2LOAD** library, which contains the **SASSBSTR** program.
8. Set **UCC7CMDS DD** to your CA7 communications dataset.
9. Set **BATCHIN DD** to your **CA7 BATCHIN** dataset.

10. Set **BATCHOUT DD** to your **CA7 BATCHOUT** dataset.
11. **SYSPRINT** must refer to the sequential dataset referenced by the symbolic **CA7PRINT** parameter, set previously in step (4). The dataset name cannot be set to the symbolic **CA7PRINT** parameter at this location because the AD Connect for Mainframe started task must extract the full dataset name while reads the JCL member. It occurs before JES substitutes the symbols. The **DSN** field must remain on the same line as **SYSPRINT DD**.
12. Change the **MASTER** user ID to a valid CA7 user, authorized to issue the *LIOB* command.

For reference, the following represents a copy of the **IAYCA7X** sample job that is contained in the **SMP/E *.SIAYSAMP** distribution library for IBM AD Connect for Mainframe:

```
//IAYCA7X  JOB NOTIFY=&SYSUID            <===Change(1)
//*******************************************************************
//*                                                                 *
//* Licensed materials - Property of IBM                            *
//* 5737-B16 Copyright IBM Corp. 2017, 2018                         *
//* All rights reserved                                             *
//* US Government users restricted rights  -  Use, duplication or   *
//* disclosure restricted by GSA ADP schedule contract with IBM Corp. *
//*                                                                 *
//*-----------------------------------------------------------------*
//* CA7 SASSBSTR BATCH TERMINAL INTERFACE JCL                       *
//*-----------------------------------------------------------------*
//* !!! THIS JCL NEEDS TO BE CUSTOMIZED BEFORE USE !!!              *
//*-----------------------------------------------------------------*
//* Please refer to file $README in the SIAYSAMP library for a      *
//* description of each customization step, '<===Change(n)'        *
//*******************************************************************
//JOBLIB  DD DISP=SHR,DSN=IAYV510.SIAYAUTH   <===Change(2)
// SET SYSINDSN='IAYV510.SAMP.JCL(CA7T1%TK)' <===Change(3)
// SET CA7PRINT='IBMUSER.CA7.REPORT.FILE%TK' <===Change(4)
//*-----------------------------------------------------------------
//* This step will call Rexx program IAYDSNDL to clean up work datasets
//* from the previous run.
//*-----------------------------------------------------------------
//STEP005 EXEC PGM=IKJEFT01,PARM='IAYDSNDL &CA7PRINT'
//SYSEXEC  DD DISP=SHR,DSN=IAYV510.SIAYSAMP          <===Change(5)
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//*-----------------------------------------------------------------
//* This step will call Rexx program IAYRXCA7 to build the CA7 control
//* cards and call CA7 report utility program SASSBSTR.
//*-----------------------------------------------------------------
//STEP010 EXEC PGM=IKJEFT01,PARM='IAYRXCA7 &SYSINDSN'
//SYSEXEC  DD DISP=SHR,DSN=IAYV510.SIAYSAMP       <===Change (6)
//*
//STEPLIB  DD DISP=SHR,DSN=CA7V12.CAL2LOAD        <===Change (7)
//*
//UCC7CMDS DD DISP=SHR,DSN=CA7V12.COMMDS          <===Change (8)
//BATCHIN  DD DISP=SHR,DSN=YOUR.CA7.BATCHI#1      <===Change (9)
//BATCHOUT DD DISP=SHR,DSN=YOUR.CA7.BATCHO#1      <===Change (10)
//SYSPRINT DD DISP=(NEW,CATLG),DSN=IBMUSER.CA7.REPORT.FILE%TK, <=Chg(11)
// SPACE=(TRK,(50,20),RLSE),DCB=(RECFM=FBA,LRECL=133)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSUDUMP DD SYSOUT=*
//*
//CA7SYSIN DD *,DLM=##  <===Change(12) (replace ID MASTER if necessary)
/LOGON MASTER
LJOB,JOB=*,LIST=ALL
/LOGOFF
##
```

Specific configurations steps for the **IAYCA712** job:

Note: Each step that is listed below is keyed by a change number (n). The sample JCL contains a change marker of the form <===Change(n) and indicates the location of the corresponding change.

1. Modify the job card to meet your site/system requirements.
2. Set **JOBLIB** to the IBM AD Connect for Mainframe **SMP/E *.SIAYAUTH** library.
3. Set **PROCS JCLLIB** to reference the **CA7 PROC LIB** where the **CA7LOG** procedure is stored.

4. Set the symbolic **SYSINDSN** parameter to a PDS/E library where this job can write a temporary CA7 report control card. Any suitable library can be used, but the member name **must** be set to **CA7T1%TK** as shown in the sample. Ensure that the IBM AD Connect for Mainframe started task has *UPDATE* access to this library.
5. Set the symbolic **CA7PRINT** parameter to a sequential dataset name. Any suitable dataset name can be user, but the final **%TK** prefix **must** be retained. The AD Connect for Mainframe started task replaces all instances of **%TK** with a unique task number (up to MAXTASK) so that each subtask can run in parallel with a unique dataset name. Ensure that the IBM AD Connect for Mainframe started task has *UPDATE* access to this sequential dataset.
6. **SYSEXEC DD** needs to point to a library, which contains the IBM AD REXX samples. You can use the IBM AD Connect for Mainframe **SMP/E *.SIAYSAMP** library if you want, since no user modifications need to be made to the REXX samples.
7. **SYSEXEC DD** needs to point to a library, which contains the IBM AD REXX samples. You can use the IBM AD Connect for Mainframe **SMP/E *.SIAYSAMP** library if you want, since no user modifications need to be made to the REXX samples.
8. **SYSPRINT** must refer to the sequential dataset referenced by the symbolic **CA7PRINT** parameter, set previously in step (4). The dataset name cannot be set to the symbolic **CA7PRINT** parameter at this location because the AD Connect for Mainframe started task must extract the full dataset name while reads the JCL member. It occurs before JES substitutes the symbols.

   The **DSN** field must remain on the same line as **SYSPRINT DD**. Other **SYSPRINT DD** fields may be moved as needed.
9. Set **UCC7HIST DD** to reference a concatenation of the following datasets:
   a. The current generation of the CA7 history dataset.
   b. The CA7 primary log dataset.
   c. The CA7 secondary log dataset.
10. Set **UCC7ARCH DD** to reference the current generation of the CA7 archive dataset.

Note: Ensure that the AD Connect for Mainframe started task has **READ** access to the CA7 datasets referenced in steps (8) and (9).

For reference, the following represents a copy of the **IAYCA712** sample job that is contained in the **SMP/E *.SIAYSAMP** distribution library for IBM AD Connect for Mainframe:

```
//IAYCA712 JOB NOTIFY=&SYSUID             <===Change(1)
//*********************************************************************
//*                                                                   *
//* Licensed materials - Property of IBM                              *
//* 5737-B16 Copyright IBM Corp. 2017, 2018                           *
//* All rights reserved                                               *
//* US Government users restricted rights  -  Use, duplication or     *
//* disclosure restricted by GSA ADP schedule contract with IBM Corp. *
//*                                                                   *
//*-------------------------------------------------------------------*
//* CA7 HISTORY REPORTING JCL -used for incremental updates           *
//*-------------------------------------------------------------------*
//* !!! THIS JCL NEEDS TO BE CUSTOMIZED BEFORE USE !!!                *
//*-------------------------------------------------------------------*
//* Please refer to file $README in the SIAYSAMP library for a        *
//* description of each customization step, '<===Change(n)'          *
//*********************************************************************
//JOBLIB  DD DISP=SHR,DSN=IAYV510.SIAYAUTH   <===Change(2)
//PROCS  JCLLIB ORDER=CA7V12.SYSGEN.JCLLIB   <===Change(3)
//*
// SET SYSINDSN='IAYV510.SAMP.JCL(CA7TC%TK)'      <===Change(4)
// SET CA7PRINT='IBMUSER.CA7.REPORT12.FILE%TK'    <===Change(5)
//*--------------------------------------------------------------------
//* This step will call Rexx program IAYDSNDL to clean up work datasets
//* from the previous run.
//*--------------------------------------------------------------------
//STEP005 EXEC PGM=IKJEFT01,PARM='IAYDSNDL &CA7PRINT'
//SYSEXEC  DD DISP=SHR,DSN=IAYV510.SIAYSAMP         <===Change(6)
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//*--------------------------------------------------------------------
```

```
//* This step will call Rexx program IAYRXCAC to build the CA7 control
//* cards.
//* Ensure that the AD Connect started task has READ access to the CA7
//* datasets referenced in steps (9) and (10).
//*--------------------------------------------------------------------
//STEP010 EXEC PGM=IKJEFT01,PARM='IAYRXCAC &SYSINDSN'
//SYSEXEC  DD DISP=SHR,DSN=IAYV510.SIAYSAMP         <===Change(7)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSPRINT DD DSN=IBMUSER.CA7.REPORT12.FILE%TK,     <===Change(8)
//            DISP=(NEW,CATLG),
//            SPACE=(TRK,(50,20),RLSE),DCB=(RECFM=FBA,LRECL=133)
//CA7SYSIN DD *
SASSHR
/*
//*--------------------------------------------------------------------
//* This step will call CA7 history report program SASSHIS8.
//*--------------------------------------------------------------------
//REPORT EXEC  CA7LOG,PG=SASSHIS8,PA='O,100000'
//SASSRA01 DD  SYSOUT=*,DCB=BLKSIZE=133
//SASSRA02 DD  SYSOUT=*,DCB=BLKSIZE=133
//SASSRA03 DD  SYSOUT=*,DCB=BLKSIZE=133
//COMMANDS DD  DSN=&&COMMANDS,DISP=(,PASS),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//             UNIT=3390,SPACE=(CYL,(1,1))
//HR25REPT DD  DSN=&&HR25REPT,DISP=(,PASS),
//             DCB=(RECFM=FB,LRECL=82,BLKSIZE=0),
//             UNIT=3390,SPACE=(TRK,(1,1),RLSE)
//HR25CSV  DD  DSN=&&HR25CSV,DISP=(,PASS),
//             DCB=(RECFM=VB,LRECL=3000,BLKSIZE=0),
//             UNIT=3390,SPACE=(TRK,(1,1),RLSE)
//UCC7HIST DD  DISP=SHR,DSN=CA7.LOG.HISTORY(+0)     <===Change(9)
//         DD  DISP=SHR,DSN=CA7V12.LOGP
//         DD  DISP=SHR,DSN=CA7V12.LOGS
//UCC7ARCH DD  DISP=SHR,
//             DSN=CA7V12.LOGARCH(+0)               <===Change(10)
//SYSLIST  DD DISP=OLD,DSN=&CA7PRINT
//*
//SYSIN DD DISP=SHR,DSN=&SYSINDSN
```

For reference, the following represents a copy of the **IAYDSNDL** sample job that is contained in the **SMP/E *.SIAYSAMP** distribution library for IBM AD Connect for Mainframe:

Note: Early builds of FMID HALT510 did not contain REXX program **IAYDSNDL**. If **IAYDSNDL** is not present in **SMP/E *.SIAYSAMP** distribution library and you require job data from CA7, create this member in the **SYSEXEC** library that is referenced in sample jobs **IAYCA7X** and **IAYCA712**.

```
/* rexx */
/*********************************************************************/
/*  IAYDSNDL                                                         */
/*  Licensed materials - Property of IBM                             */
/*  5737-B16 Copyright IBM Corp. 2017, 2018                          */
/*  All rights reserved                                              */
/*  US Government users restricted rights  -  Use, duplication or    */
/*  disclosure restricted by GSA ADP schedule contract with IBM Corp. */
/*                                                                   */
/*********************************************************************/

arg dsname

   dsname = strip(dsname)
   upper dsname
   if dsname = '' then
      return 7

   xx = SYSDSN("'"dsname"'")
   if xx = 'OK' then
      "DELETE ('"dsname"') "
   return 0

exit 0
```

Configuration for Db2

IBM AD Connect for Mainframe supports all currently supported versions of Db2 for z/OS. At the time of this publication, it includes the minimum supported version **11.1** and higher.

Previous versions of IBM AD Connect for Mainframe that allowed for the creation of DBRM-based plans on Db2 are no longer supported. You need to create the Db2 packages and a plan if you want to allow IBM AD Connect for Mainframe to gather Db2 operational information as part of your AD/ADDI project. For more information, perform the following steps:

Note:

- Make sure to use the same package name to create the packages and plan. The package name appears in sample job IAYBNDPL under PKLIST. If you want to change the package name, update both sample jobs IAYBNDPK and IAYBNKPL.
- The owner of the packages and plan must have the authority to run the **BIND** command on the specified PKLIST.
- The **name of the plan** that is used to access Db2 from IBM AD Connect for Mainframe must be **IAYSRC**.

1. Use sample job IAYBNDPK to create packages. Modify the sample according to your environment.

```
//IAYBNDPK JOB <job parameters>
//JOBLIB DD DSN=DSN1210.SDSNLOAD,DISP=SHR <=== DB2 Loadlib
//BINDUNL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBCG) <==set owner (see all cases of owner in the job)
  BIND PACKAGE (IAY60PKG) OWNER(userdefined) +  <== set owner (see all cases of owner in the
job)
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2DB) LIBRARY('IAYV600.SIAYDBRM')  <== DBRM library (same goes for all
appearance in job)
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2VL) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2TB) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2TS) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2PL) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2CL) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2SG) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2IX) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2KY) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2TG) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2VW) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2PK) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2PX) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
     MEMBER(IAYDB2RT) LIBRARY('IAYV600.SIAYDBRM')
  BIND PACKAGE (IAY60PKG) OWNER(IBMUSER)                +
     ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)     +
  MEMBER(IAYDB2PR) LIBRARY('IAYV600.SIAYDBRM')
  END
//
```

Figure 6. Sample job IAYBNDPK

2. Use sample job IAYBNDPL to create a plan. The plan must refer to the packages that are created in the previous step. Modify the sample according to your environment.

```
//IAYBNDPL JOB <job parameters>
//*
//JOBLIB DD DSN=DSN1210.SDSNLOAD,DISP=SHR <== DB2 Loadlib
//BINDUNL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
 DSN SYSTEM(DBCG)  <=== DB2 Subsystem name
 BIND PLAN(IAYSRC) OWNER(userdefined) + <== set owner (see all   cases of owner in the job)
 ACTION(ADD) ISOLATION(CS) +
 PKLIST(IAY60PKG.*)
 END
//SYSIN DD *
GRANT EXECUTE ON PLAN IAYSRC TO PUBLIC;
//*
//
```

Figure 7. Sample job IAYBNDPL

Configuration for IBM MQ

1. Confirm that the IBM MQ **CSQAUTH** and **CSQLOAD** load libraries are added to the IBM AD Connect for Mainframe started task STEPLIB as outlined in the “Configuring the Listener PROC” on page 152 section.
2. Confirm that IBM MQ queue **SYSTEM.COMMAND.INPUT** is defined on the LPAR where IBM AD Connect for Mainframe is running. This queue is part of the default IBM MQ installation on z/OS.
3. Define a local IBM MQ command reply queue. It can be any queue with attributes that allow reply messages to be put on it. It must be named **IAYEGACY.COMMAND.REPLY**.

IBM AD ParmLib

If ChangeMan ZMF, Adabas, or Natural is to be used, a PDS/E library should be created. This library would be referenced for now on as the IBM AD ParmLib and does not have to be a PROCLIB. The IBM AD ParmLib will contain the customized JCL, based on the samples in the SIAYSAMP SMP/E library, that allows the IBM AD Connect for Mainframe started task to interface with ChangeMan ZMF, Adabas, or Natural.

In IBM AD Build Configuration Admin, the IBM AD ParmLib should be specified accordingly.

Configuration for Adabas

When **IBM AD Connect for Mainframe** is required to access **Adabas**, the following sample job needs to be customized. The sample job is found in the **SMP/E *.SIAYSAMP** library, more exactly in the **IAYNATDB** member. In **IBM AD ParmLib** library, the member name must be **NATDBJOB**, although the job name can be altered.

Important:

- The job is used by IBM AD Connect for Mainframe. Therefore, it shouldn’t be submitted manually.
- Make sure that this job is running on the **same** LPAR as the IBM AD Connect for Mainframe started task. If your shop is configured in such a way that the job might be scheduled on another system, make sure that you use the **SYSTEM=*** parameter in the job card, as shown in the example. The ***** indicates the system that submitted the job. It does not represent a wildcard substitution for any system.
- Follow the customization steps for the **IAYNATDB** job in the comment section of the sample JCL.

```
//IAYNATDB JOB <job parameters>,SYSTEM=* <== run job on same system as AD
//*
//REP EXEC PGM=ADARUN   <==Modify to match your site standard
//STEPLIB DD DISP=SHR,DSN=ADABAS.ADAvrs.LOAD <=== ADABAS LOAD
//        DD DISP=SHR,DSN=ADABAS.APSvrs.LDnn <=== APS load update
//        DD DISP=SHR,DSN=ADABAS.APSvrs.LD00 <=== APS load base
//*
```

```
//DDASSOR1  DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1  DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1  DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDSAVE    DD   DISP=SHR,DSN=EXAMPLE.DByyyyy.SAVE   <=== SAVE DATASET
//*
//DDDRUCK DD DSN=IAY.NATURAL.DB.TASK%TKK,DISP=SHR <=the DSN for IAYDR0nn  ..
//*                              ...DD names allocated by the AD Connect STC.
//*                              The %TKK suffix must remain.
//DDCARD DD *
ADARUN PROG=ADAREP,SVC=%SVC,DEVICE=%DEV,DBID=%DBI
/*
//DDKARTE DD *
ADAREP REPORT
/*
//    IF   (ABEND=TRUE|ABEND=FALSE) THEN
//POST EXEC PGM=IAYPOST
//STEPLIB  DD DISP=SHR,DSN=IAYV510.SIAYAUTH <== Set to your AD loadlib
//SYSUDUMP DD SYSOUT=*
//INDCB DD *
//     ENDIF
/*
```

In addition, create datasets called IAY.NATURAL.DB.TASKnn as sequential files with record format FB and record length 133, where nn goes from 01 to the number of tasks specified in **MAXTASK**.

Configuration for Natural

The following 2 jobs should be created in **IBM AD ParmLib** library.

Important: The jobs are used by **IBM AD Connect for Mainframe**. Therefore, they shouldn't be submitted manually.

1. The following sample job is found in the **SMP/E *.SIAYSAMP** library, more exactly in the **IAYNATLB** member. Follow the customization steps for the **IAYNATLB** job listed in the comment section of the sample JCL. After customization, the member name needs to be set to **NATLBJOB**, although the job name can be altered. Setting the member name to **NATLBJOB** it allows the **IBM AD Connect for Mainframe** to locate the correct job to retrieve the list of members or DDM views.

   ```
   //IAYNATLB JOB <job parameters>,SYSTEM=* <== run job on same system as AD
   //NAT EXEC PROC=NATB003   <== Substitute your site's Natural batch procedure
   //*
   //CMPRINT DD DSN=IAY.NATURAL.LB.TASK%TKK,DISP=SHR <- the DSN for IAYCA0nn ...
   //*                              ...DD names allocated by the AD Connect STC.
   //*                              The %TKK suffix must remain.
   //CMSYNIN DD *
   LOGON %LIB
   L %REQ
   FIN
   /*
   //*
   //    IF   (ABEND=TRUE|ABEND=FALSE) THEN
   //POST EXEC PGM=IAYPOST
   //STEPLIB  DD DISP=SHR,DSN=IAYV510.SIAYAUTH <== Set to your AD loadlib
   //SYSUDUMP DD SYSOUT=*
   //INDCB DD *
   //     ENDIF
   /*
   ```

   In addition, create datasets called IAY.NATURAL.LB.TASKnn as sequential files with record format FB and record length 133, where nn goes from 01 to the number of tasks specified in **MAXTASK**.

2. The following sample job is found in the **SMP/E *.SIAYSAMP** library, more exactly in the **IAYNATLB** member. Follow the customization steps for the **IAYNATLB** job listed in the comment section of the sample JCL. After customization, the member name needs to be set to **NATMMJOB**, although the job name can be altered. Setting the member name to **NATMMBJOB** it allows the **IBM AD Connect for Mainframe** to locate the correct job to retrieve the member source records.

   ```
   //IAYNATMM JOB <job parameters>,SYSTEM=* <== run job on same system as AD
   //NAT EXEC PROC=NATB003    <== Substitute your site's Natural batch procedure
   //*
   //CMPRINT DD DSN=IAY.NATURAL.MM.TASK%TKK,DISP=SHR -< the DSN for IAYCB0nn
   //*                              ...DD names allocated by the AD Connect STC.
   //*                              The %TKK suffix must remain.
   //CMSYNIN DD *
   LOGON %LIB
   L %MEM
   FIN
   /*
   //*
   //    IF   (ABEND=TRUE|ABEND=FALSE) THEN
   //POST EXEC PGM=IAYPOST
   //STEPLIB  DD DISP=SHR,DSN=IAYV510.SIAYAUTH <== Set to your AD loadlib
   //SYSUDUMP DD SYSOUT=*
   //INDCB DD *
   //     ENDIF
   /*
   ```

In addition, create datasets called IAY.NATURAL.MM.TASKnn as sequential files with record format FB and record length 133, where nn goes from 01 to the number of tasks specified in **MAXTASK**.

Configuration for ChangeMan ZMF

The IBM AD Connect for Mainframe started task retrieves source files from **ChangeMan ZMF** by submitting a user configured batch job through the internal reader. The sample JCL for this batch job is provided in the **IAYXMLRQ** member of the **SMP/E *.SIAYSAMP** distribution library.

The following job needs to be created whenever accessing **ChangeMan ZMF** via XML service.

The job needs to be in a **PDS/E**, member name must be **XMLREQJB**, job name can be altered, and the library doesn't have to be a **PROCLIB**.

Note:

- The job is used by IBM AD Connect for Mainframe. Therefore, it shouldn’t be submitted manually.
- Make sure that this job is running on the **same** LPAR as the AD Connect for Mainframe started task. If your shop is configured in such a way that the job might be scheduled on another system, make sure that you use the **SYSTEM=*** parameter in the job card, as shown in the example. The ***** indicates the system that submitted the job. It does not represent a wildcard substitution for any system.

```
//IAYCHGXL JOB <job parameters>
//*********************************************************************
//* Licensed materials - Property of IBM                              *
//* 5737-B16 Copyright IBM Corp. 2017, 2018                           *
//* All rights reserved                                               *
//* US Government users restricted rights  -  Use, duplication or     *
//* disclosure restricted by GSA ADP schedule contract with IBM Corp. *
//*                                                                   *
//* ChangeMan ZMF Sample JCL.                                         *
//*                                                                   *
//* Customize for your site as follows:                               *
//*                                                                   *
//* 1) Add the job parameters to meet your system requirements.       *
//*                                                                   *
//* 2) Set STEPLIB to point to :                                      *
//*        -Changeman User Loadlib                                    *
//*        -Changeman SERCOMC loadlib                                 *
//*        -Changeman CMNZMF  loadlib                                 *
//*        -IBM AD loadlib (SIAYAUTH)                                 *
//*                                                                   *
//* 3) Set the XMLIN and XMLOUT dataset names to match those          *
//*    that have been pre-allocated as part of the AD Connector       *
//*    started task configuration.  For this job, DO NOT include      *
//*    the 2 trailing digits in the pre-allocated dataset names,      *
//*    since the AD connector task will insert them before            *
//*    submitting the job to the internal reader.                     *
//*                                                                   *
//* 4) In the POST step, set the STEPLIB to the                       *
//*    IBM AD loadlib (SIAYAUTH)                                      *
//*                                                                   *
//* See the IBM AD Connect for Mainframe Configuration Guide          *
//* for more details.                                                 *
//*********************************************************************
//*
//XML EXEC PGM=SERXMLBC
//STEPLIB DD DISP=SHR,DSN=CHGMAN.USER.LOAD <== Changman User loadlib
// DD DISP=SHR,DSN=CHGMAN.SERCOMC.LOAD    <== Changeman SERCOMC library
// DD DISP=SHR,DSN=CHGMAN.CMNZMF.LOAD     <== Changeman CMNZMF library
// DD DISP=SHR,DSN=IAYV510.SIAYAUTH       <== IBM AD load library
//SER#PARM DD DISP=SHR,DSN=CHGMAN.SERCOMC.TCPIPORT <== Change ZMF
//*                                                    TCPIPORT File
//SYSPRINT DD SYSOUT=Z
//SERPRINT DD SYSOUT=Z
//SERPRINT DD SYSOUT=Z
//SYSOUT DD SYSOUT=Z
//XMLIN DD DISP=SHR,DSN=IAY.CHGMAN.XMLIN <=== The input PDS as defined
//*                                          in the Listener job
//XMLOUT DD DISP=SHR,DSN=IAY.CHGMAN.XMLOUT <=== The output of XML processing
//*                                  as defined in the Listener job
// IF (ABEND=TRUE|ABEND=FALSE) THEN
//POST EXEC PGM=IAYPOST,COND=(16,LT,XML)
//STEPLIB DD DISP=SHR,DSN=IAYV510.SIAYAUTH   <== Set to AD load lib
```

```
//SYSUDUMP DD SYSOUT=*
//INDCB DD *
// ENDIF
```

Note: Make sure that the ChangeMan ZMF batch jobs are running on the same LPAR where **IBM AD Connect for Mainframe** is running.

In addition, create datasets that are named IAY.CHGMAN.XMLINnn as sequential files with record format VB and record length 255, where nn goes from 01 to the number of tasks specified in MAXTASK.

Create datasets that are named IAY.CHGMAN.XMLOUTnn as sequential files with record format VB and record length 5000, where nn goes from 01 to the number of tasks specified on MAXTASK.

The initial space allocation (in units of tracks) for the above datasets is based on the following formula: **Take the number of members in your largest ChangeMan library and divide by 80, then add an additional 10%** . For example, if your largest ChangeMan library contains 44000 members, then **44000/80 + 10% = 550+55 = 605** tracks.

Ensure that each of these newly defined datasets is correctly referenced on the XMLINnn and XMLOUTnn DD names in the IAYLSTNR PROC previously customized. There is a 1 to 1 correspondence between each dataset and XML* DD name in **IAYLSTNR**.

Note: Prior to v5.1.0.2 release of **IBM AD Connect for Mainframe**, the XMLIN and XMLOUT dataset names that were referenced in **IAYXMLRQ** had to be a specific length. This was due to a restriction in the code that appended the subtask number to the dataset name at a fixed offset. The last character of the XMLIN dataset name had to be on column 43 and the last character of the XMLOUT dataset name had to be on column 44. Starting with the v5.1.0.2 release of **IBM AD Connect for Mainframe**, this restriction has been removed. The dataset name can be any valid length and the final character can occur at any column offset that is valid within the syntax rules for JCL.

For Continuous Rule Validation via ChangeMan integration, the following step must be added in the compile skeleton(s) in ChangeMan:

```
//RUNTEST EXEC PGM=IAYCLINT,PARM='&SUBSYS,&CMPNAME,&CMPTYPE,&PKGNAME,<nn>,&SYSUID',
// REGION=0M
//STEPLIB DD DSN=IAYV502.LOADLIB,DISP=SHR <== IBM AD load library
//COMPN DD DISP=(OLD,PASS),DSN=&&LIST <== ChangeMan Listing Report output
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//INDCB DD *
<xxx.xxx.xxx.xxx>< ><ppppp>
```

Where <nn> is the return code in case the validation couldn't be launched, *<xxx.xxx.xxx.xxx>* is the IP address of the IBM AD Validation server and *<ppppp>* is the port.

Enabling TLS Connection to IBM AD Validation Server

Before the Application Transparent Transport Layer Security (AT-TLS) setup in IBM AD Connect for Mainframe, make sure to complete the enablement of Transport Layer Security (TLS) connection in IBM AD Validation Server. For instructions, see "Enabling TLS Connection between IBM AD Validation Server and IBM AD Connect for Mainframe" on page 290.

For IBM AD V5.1.0.7 and earlier versions, the communication between IBM AD Validation Server and IBM AD Connect for Mainframe is unencrypted socket session. Beginning with version 5.1.0.8, the optional secure communication, which uses the TLS protocol, is supported by using the AT-TLS feature of IBM z/OS Communication Server.

The TLS protocol is a client or server cryptographic protocol. It is based on the earlier Secure Sockets Layer (SSL) specifications that are developed by Netscape Corporation for securing communications that use Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. The TLS and SSL protocols are designed to run at the application level. Therefore, typically, an application must be designed and coded to use TLS/SSL protection. On z/OS, the System SSL component of the Cryptographic Services element implements the full suite of SSL and TLS protocols (SSL V2, SSL V3, TLS V1.0, TLS V1.1, and TLS V1.2 as of this writing), including a robust set of application programming interfaces (APIs) for z/OS C and C++ applications to use.

To make the TLS or SSL protocol more accessible to z/OS applications, z/OS Communications Server V1R7 introduced the AT-TLS feature. AT-TLS starts TLS or SSL primitives in the TCP layer of the TCP/IP stack on behalf of application programs, based on policy files that describe the application traffic and how to protect it. With AT-TLS, z/OS applications that are written in almost any language can enjoy full TLS or SSL protection without requiring source code changes. AT-TLS allows socket applications to access encrypted sessions by invoking system SSL within the transport layer of the TCP/IP stack.

AT-TLS policy is read, parsed, and installed into the TCP/IP stack by the z/OS Communication Server Policy Agent (PAGENT), which implements policy-based networking for the z/OS environment. The application continues to send and receive clear text over the socket, but data that is sent over the network is protected by the System SSL component. For more information about policy-based networking, see *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

To configure IBM AD Connect for Mainframe as a server with TLS/SSL support, perform the following steps. If you have already configured AT-TLS for other servers on your z/OS system, begin at step 5.

1. Configure PAGENT as a started task on z/OS.
2. Define the security authorization for PAGENT.
3. Define PAGENT configuration files.
4. Configure AT-TLS.
5. Store the personal certificate that represent IBM AD Validation Server.
6. Import the personal certificate to the RACF database.
7. Set up the PAGENT policy for the port that is used by IBM AD Validation Server.
8. Refresh the PAGENT policy.

1. Configure PAGENT as a started task on z/OS.

   PAGENT runs as a UNIX process, so it can be started either from the UNIX System Service shell or as a z/OS started task. The following example shows how to start PAGENT as a z/OS started task. You can also find this sample file in TCPIP.SEZAINST(EZAPAGSP).

```
//PAGENT PROC
//PAGENT EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//STDENV DD DSN=SYS1TCPPARMS(PAGENTV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

   Figure 8. Example: PAGENT JCL started procedure

   To run with the required configuration, you can use environment variables that are configured either in an IBM Multiple Virtual Storage (IBM MVS™) data set, or in a z/OS UNIX file that is specified by the STDENV data definition (DD). The following example shows how to configure the environment variables for PAGENT. You can also find this sample file in SYS1.TCPPARMS(PAGENTV).

```
TZ=PST8PDT7
PAGENT_CONFIG_FILE=//'SYS1.TCPPARMS(PAGENTP)' == > in unix default file is /etc/pagent.conf
PAGENT_LOG_FILE=SYSLOGD == > it could be /tmp/pagent.log
```

   Figure 9. Example: SYS1.TCPPARMS(PAGENTV) data set containing PAGENT environment variables

   TZ
   Specifies the local time zone for the PAGENT process. In the preceding example, it is set to Pacific Standard Time (GMT -8) and Pacific Daylight Saving Time (GMT -7).

   PAGENT_CONFIG_FILE
   Specifies the PAGENT configuration file. In the preceding example, the PAGENT configuration file is a member of the SYS1.TCPPARMS data set. For more information about PAGENT configuration files, see step 3.

PAGENT_LOG_FILE
Specifies the name of the log file that is used by PAGENT. In the preceding example, it is set to log PAGENT messages into file SYSLOGD.

Before you start the PAGENT started task, make sure that the appropriate security authorizations have been set. See the following step for details.

2. Define the security authorization for PAGENT.

   The policies that are managed by PAGENT can affect system operation significantly. Therefore, you must restrict the access of z/OS user IDs to PAGENT. To do this, define certain resources and controls in the security manager product of the system, for example, RACF. To set up the security definitions of PAGENT to RACF, follow the steps:

   a. Define the PAGENT started task to RACF.

      To set up the PAGENT started task to RACF, you must define a profile to the STARTED RACF generic resource class by using the **RDEFINE** command. The following example shows the RACF commands that are used to set up the PAGENT started task. You can also find this sample file in SIAYSAMP(RACFPAG1).

```
//DAEMONS EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(STARTED)
SETROPTS RACLIST(STARTED)
SETROPTS GENERIC(STARTED)
RDEFINE STARTED PAGENT.*
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
/*
```

      Figure 10. Example: RACF commands to define the PAGENT started task to RACF

      If you also want to log messages into the SYSLOGD file that is mentioned in Figure 9 on page 165, use the **RDEFINE** command to define a profile for SYSLOGD to the STARTED class, for example, `RDEFINE STARTED SYSLOGD.*`.

   b. Define the PAGENT user ID.

      Before z/OS V1R13, the user ID for the PAGENT started task must have the z/OS UNIX superuser authority, which means the z/OS UNIX user ID (UID) for this user must be set to 0, and you must assign a default group (DFLTGRP) for the user ID. In the following example, PAGENT, the policy agent, runs under the z/OS user ID PAGENT. The RACF command is used to define the PAGENT user ID to the OMVSGRP default group with an OMVS segment that has a UID of 0. You can also find this sample file in SIAYSAMP(RACFPAG2).

```
/PAUSER EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ADDUSER PAGENT DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
/*
```

      Figure 11. Example: RACF command to define a user ID for the PAGENT started task

      If the SHARED.IDS profile has been defined in the UNIXPRV class, the UID value must be unique. If you want to use a UID value that is already in use, you can indicate that you intend to share the UID value across z/OS user IDs by adding the SHARED keyword to the UID parameter of the **ADDUSER** command.

      If you also want to log messages into the SYSLOGD file that is mentioned in Figure 9 on page 165, use the **ADDUSER** command to define a user ID with the superuser authority for the SYSLOGD started task, for example, `ADDUSER SYSLOGD DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))`.

      If you are using z/OS V1R13 or later, you can choose to use a non-superuser UID. For more information about using such a UID, see *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

c. Associate the PAGENT user ID with the PAGENT started task.

Use the RACF **RALTER** command to associate the PAGENT user ID, which is created from the preceding step, to the PAGENT started task. See the following example. You can also find this sample file in SIAYSAMP(RACFPAG3).

```
//ASCPAUSR EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RALTER STARTED PAGENT.* STDATA(USER(PAGENT))
/*
```

Figure 12. Example: RACF command to associate the PAGENT user ID with the PAGENT started task

If you also want to log messages into the SYSLOGD file that is mentioned in Figure 9 on page 165, use the **RALTER** command to associate the SYSLOGD user ID with the SYSLOGD started task, for example, `RALTER STARTED PAGENT.* STDATA(USER(SYSLOGD))`.

d. Give authorized users access to manage the PAGENT started task.

To restrict management access to the PAGENT started task, you must define a profile that is named MVS.SERVMGR.PAGENT in the OPERCMDS RACF resource class, and give only authorized users access to this facility. The following example shows the RACF commands that are used to configure this access. You can also find this sample file in SIAYSAMP(RACFPAG4).

```
//PERMITPA EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(OPERCMDS)
SETROPTS RACLIST (OPERCMDS)
RDEFINE OPERCMDS (MVS.SERVMGR.PAGENT) UACC(NONE)
PERMIT MVS.SERVMGR.PAGENT CLASS(OPERCMDS) ACCESS(CONTROL) -
ID(PAGENT)
SETROPTS RACLIST(OPERCMDS) REFRESH
/*
```

Figure 13. Example: RACF commands to give authorized users access to manage the PAGENT started task

e. Restrict access to the **pasearch** UNIX command.

The **pasearch** UNIX command can be used to display policy definitions. The output from this command indicates whether policy rules are active and shows the parsed results of the policy definition attributes. PAGENT, the policy agent is designed to ignore unknown attributes, so misspelled attributes result in the default values being used.

The **pasearch** output can be used to verify whether policies are defined correctly. However, you might not want every user to be able to see the policy definitions. To restrict unauthorized access to the **pasearch** command, a resource is defined to the SERVAUTH RACF resource class. This type of resource can be defined for each TCP/IP stack (TcpImage) and policy type (ptype) by using the following command:

```
EZB.PAGENT.sysname.TcpImage.ptype
```

sysname
The z/OS system name.

TcpImage
The procedure name of the TCP/IP stack that policy information is to be restricted to.

ptype
The policy type that is being requested. The following types are possible:

QoS
Policy Quality of Service

IDS
Policy Intrusion Detection System

IPSec
Policy Internet Protocol Security

TTLS
Policy AT-TLS

The following example shows the RACF commands that are used to restrict access to the **pasearch** UNIX command. In this example, only the USRT001 and USRT002 z/OS user IDs are permitted to use the **pasearch** command for the TCPIP TCP/IP stack on the UTEC224 z/OS system. You can also find this sample file in SIAYSAMP(RACFPAG4).

```
//PAGNACC EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RDEFINE SERVAUTH EZB.PAGENT.UTEC224.TCPIP.* UACC(NONE)
PERMIT EZB.PAGENT.UTEC224.TCPIP.* CLASS(SERVAUTH) -
ID(USRT001) ACCESS(READ)
PERMIT EZB.PAGENT.UTEC224.TCPIP.* CLASS(SERVAUTH) -
ID(USRT002) ACCESS(READ)
SETROPTS GENERIC(SERVAUTH) REFRESH
SETROPTS RACLIST(SERVAUTH) REFRESH
/*
```

*Figure 14. Example: RACF commands to restrict unauthorized users access to the **pasearch** UNIX command*

3. Define PAGENT configuration files.

   PAGENT is responsible for reading policies from configuration members in MVS data sets or in z/OS UNIX file system files. Before you can define AT-TLS policies, you must configure certain operational characteristics of PAGENT in a main configuration file. The following statements can be contained in the main configuration file:

   - The TcpImage statement
   - The LogLevel statement

   Define the TcpImage statements and appropriate logging level in the main configuration file.
   The following example shows definitions in the main configuration file for a single TCP/IP stack environment. You can also find this sample file in UNIX default file `/etc/pagent.conf` or in SIAYSAMP(CNFGPAG). In this example, the AT-TLS policies are defined in a separate image-specific configuration file, TTLSPOL, for the TCPIP TCP/IP image.

```
# LogLevel statement
# SYSERR, OBJERR, PROTERR, and WARNING messages are logged.
LogLevel 15
#
# TcpImage statement
# TCP/IP image: TCPIP
# Path to image-specific configuration file: SYS1.TCPPARMS(TTLSPOL)
# FLUSH parameter specified to delete existing policy data in the
# stack on PAGENT start-up or when the configuration files change.
# PURGE parameter specified to delete active policy data from the
# stack and Policy Agent when PAGENT is shut down normally.
TcpImage TCPIP //'SYS1.TCPPARMS(TTLSPOL)' FLUSH PURGE
# Here is an alternate way to specify a common AT-TLS policy for the
# TCPIP image:
#
# TcpImage TCPIP FLUSH PURGE
# TTLSConfig //'SYS1.TCPPARMS(TTLSPOL)' FLUSH PURGE
#
```

Figure 15. Example: PAGENT main configuration file

   As mentioned in Figure 9 on page 165, `/etc/pagent.conf` is copied to SYS1.TCPPARMS(PAGENTP), and this configuration file is specified by using the **PAGENT_CONFIG_FILE** environment variable in SYS1.TCPPARMS(PAGENTV).

4. Configure AT-TLS.

AT-TLS support is enabled by specifying the TTLS parameter on the TCPCONFIG statement in the TCP/IP profile data set. The information that is required to negotiate secure connections is provided to the TCP/IP stack by the AT-TLS policies that are read, parsed, and installed by PAGENT.

When AT-TLS is enabled and a new TCP connection is established, the AT-TLS component in the TCP layer of the stack searches for an AT-TLS rule in the policy that matches the characteristics, including local and remote IP addresses, local and remote ports, the connection direction, and other characteristics. If such a rule is found, TLS protection is applied to the rule according to the details that are specified in the AT-TLS action that is associated with the rule. If no such a rule is found, the connection is not protected with TLS.

Tip: To enable TTLS without modifying PROFILE.TCPIP, and without stopping and starting the TCP/IP stack, define a separate file that contains the TCPCONFIG TTLS statement, and run the VARY TCPIP OBEYFILE command, as shown in the following example:

```
1. OBEYFILE called TTLSON in the SYS1.TCPPARMS data set:
TCPCONFIG TTLS

2. On the MVS console, issue this command:
VARY TCPIP,,O,DSN=SYS1.TCPPARMS(TTLSON)

To disable TTLS, perform the following actions:

1. OBEYFILE called TTLSOFF in the SYS1.TCPPARMS data set:
TCPCONFIG NOTTLS

2. On the MVS console, issue this command:
VARY TCPIP,,O,DSN=SYS1.TCPPARMS(TTLSOFF)
```

To make permanent changes for TLS support, update the TCP/IP configuration in PARMLIB:

- Add the TTLS parameter to the TCPCONFIG section.
- Add the PAGENT parameter to the AUTOLOG section.

Optionally, you can set up TTLS stack initialization access control for AT-TLS. When AT-TLS is started during TCP/IP stack initialization, there might be a delay between the time that the stack comes up and when PAGENT successfully installs policy information into the stack. This situation can leave a window of time where the connections that are intended to be protected by AT-TLS can be established without that protection. While this window is open, to prevent such connections from being established, define a profile for the EZB.INITSTACK resource in the SERVAUTH RACF resource class. This type of resource can be defined for each TCP/IP stack by using the following command:

```
EZB.INITSTACK.sysname.TcpImage
```

sysname
The system name that is assigned to the z/OS LPAR.

TcpImage
The procedure name of the TCP/IP stack to which access is to be controlled.

The following example shows how to set up TCP/IP stack initialization access control for AT-TLS in the profile. You can also find this sample file in SIAYSAMP(STKINIT).

```
//STACKACC EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(SERVAUTH)
SETROPTS RACLIST (SERVAUTH)
SETROPTS GENERIC (SERVAUTH)
RDEFINE SERVAUTH EZB.INITSTACK.UTEC224.TCPIP UACC(NONE)
PERMIT EZB.INITSTACK.UTEC224.TCPIP CLASS(SERVAUTH) -
ID(OMVSKERN) ACCESS(READ)
PERMIT EZB.INITSTACK.UTEC224.TCPIP CLASS(SERVAUTH) -
ID(PAGENT) ACCESS(READ)
PERMIT EZB.INITSTACK.UTEC224.TCPIP CLASS(SERVAUTH) -
ID(SYSLOGD) ACCESS(READ)
SETROPTS GENERIC(SERVAUTH) REFRESH
SETROPTS RACLIST(SERVAUTH) REFRESH
/*
```

Figure 16. Example: RACF commands to set up TCP/IP stack initialization access control for AT-TLS

With such a profile in place, you can control which applications are allowed to establish TCP connections before PAGENT is started. To do so, provide the READ access to the EZB.INITSTACK.*sysname*.*TcpImage* profile in the SERVAUTH class for the z/OS user IDs under which such applications will run. Before connections are enabled, all other applications are forced to wait until PAGENT is initialized and its policies are installed into the stack. For more information, see *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

5. As part of the TLS configuration process in IBM AD Validation Server, a personal certificate to present IBM AD Validation Server is created. This personal certificate must be signed with a certification authority (CA) certificate or a self-signed certificate. Save a copy of this CA certificate or self-signed certificate on the mainframe.

   Ensure that the certificate file is downloaded from IBM AD Validation Server in the correct format. The example in the configuration instructions for IBM AD Validation Server creates a DER encoded X.509 certificate that is in the Base64 format, so download this certificate in the ASCII mode if using FTP. An alternative to FTP is to copy and paste the certificate contents. The mainframe data set that contains the certificate must be VB with a LRECL of at least 256 bytes.

   Important: The personal certificate that is presented to IBM AD Connect for Mainframe on behalf of the IBM AD Validation Server must have the Common Name field equal to the numeric IP address of the Windows machine where the IBM AD Validation Server is running. Do not use the host name. Confirm this with the person who performs TLS configurations in IBM AD Validation Server.

6. Import the personal certificated that is saved in the preceding step to the RACF database as TRUSTED. Create an RACF key ring in the RACF database for the user ID that the skeleton in ChangeMan will run with. Connect the imported personal certificate to the key ring.

   The following example shows how to import a CA certificate to the RACF database, create a key ring, and connect a user ID to the key ring. The user ID needs the read access to IRR.DIGTCERT.LISTRING in the FACILTY class. You can also find this sample file in IAYV510.SIAYSAMP(VCERTJCL) or IAYV510.SAMPLIB(VCERTJCL).

```
//jobname JOB 'SAMPLE JCL', MSGLEVEL=1,MSGCLASS=H,CLASS=A,REGION=0M
//
*********************************************************************************************
*******
//* JOB INSTRUCTIONS: *
//* ENTER VALID JOBCARD ABOVE. *
//*1. CHANGE ++CERTIFICATE-DATASET++ TO THE SEQUENTIAL FILE CONTAINING
//*     CA CERT OR SELF SIGNED CERTIFICATE.
//* 2. CHANGE ++LABEL ++ TO APPROPRIATE LABEL NAME .
//* 3. CHANGE ++RINGNAME++ TO YOUR KEY RING NAME. EX: VALRING *
//* 4. CHANGE ++USERID++ TO THE USERID SUBMITTING THE
//*    COMPILE SKELETON(S) IN CHANGEMAN
//
*********************************************************************************************
*******
//* AT THE END OF THIS JOB YOU SHOULD HAVE THE FOLLOWING: *
//* KEYRING : ++RINGNAME++ *
//* CA CERT/SELF SIGNED CERT CONNECTED TO THE RING OWND BY USERID
//
*********************************************************************************************
*******
//*      * ADD CA CERT/SELF-SIGNED CERTIFICATE TO THE RACF DATABASE
//*      * CERTAUTH - CERTIFICATE AUTHORITY TYPE.
//
*********************************************************************************************
*******
//
*********************************************************************************************
*******
//STEP1  EXEC PGM=IKJEFT01,REGION=512K
//SYSPRINT    DD    SYSOUT=*
//SYSOUT      DD    SYSOUT=*
//SYSTSPRT    DD    SYSOUT=*
//SYSTSIN     DD    *
  RACDCERT CERTAUTH ADD('++CERTIFICATE DATASET++') +
   WITHLABEL('++LABEL++')  TRUST
  SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
 SETROPTS RACLIST(FACILITY) REFRESH
/*
//
*********************************************************************************************
//*Create RING and Connect CA cert to it.*********************************************
//*RING to be owned by USERID submitting Changeman skeleton Job.**************
//
*********************************************************************************************
//STEP2     EXEC  PGM=IKJEFT01,REGION=512K
//SYSPRINT    DD    SYSOUT=*
//SYSOUT      DD    SYSOUT=*
//SYSTSPRT    DD    SYSOUT=*
//SYSTSIN     DD    *
  RACDCERT ID(++USERID++) ADDRING(++RINGNAME++)
  RACDCERT ID(++USERID++) CONNECT(CERTAUTH LABEL('++LABEL++') +
  RING(++RINGNAME++)    +
  default
   SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
  SETROPTS RACLIST(FACILITY)
REFRESH
/*
//
*********************************************************************************************
*****
//*PERMIT USERID to access the RING
//*REFRESH CLASSES.
//
*********************************************************************************************
*****
//PERMIT      EXEC  PGM=IKJEFT01,REGION=512K
//SYSPRINT    DD    SYSOUT=*
//SYSOUT      DD    SYSOUT=*
//SYSTSPRT    DD    SYSOUT=*
//SYSTSIN     DD    *
 PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(++USERID++) ACCESS(READ)
 SETR RACLIST (FACILITY) REFRESH
/*
```

Figure 17. Example: Sample JCL to import a CA certificate to the RACF database, create a key ring, and connect a user ID to the key ring

7. Set up the PAGENT policy for the port that is used by IBM AD Validation Server.

Because IBM AD Validation Server runs on the Windows machine, and the batch job runs on the mainframe as the client, the PAGENT policy for the secured port needs to be configured for the outbound direction. The following example shows how to set up the PAGENT policy. You can also find this sample file in IAYV510.SIAYSAMP(VALTTLS).

```
TTLSRule                        VAL_SERVER
{
 RemotePortRange          ++port-number++
 Direction                Outbound
 TTLSGroupActionRef       grp_Enabled
 TTLSEnvironmentActionRef  AD_VALIDATE
}
##
}
##
TTLSGroupAction               grp_Enabled
{
 TTLSEnabled                On
 Trace                      255   # Errors to syslogd & IP joblog
#Trace                      254   # Everything to syslogd
}
##----------------------------
TTLSEnvironmentAction         AD_VALIDATE
{
 HandshakeRole Client
 TTLSKeyRingParms
 {
  Keyring ++RING-NAME++        # Keyring must be owned by the userid with which Changeman
skeleton
                                Job will run
 }
 TTLSCipherParmsRef      TLS_12 # Ciphers modern acceptable
##
 TTLSEnvironmentAdvancedParms
 {
  ApplicationControlled Off
## TSLV1.2 only on z/OS 2.1
  TLSV1.2 On               # latest tls only
  TLSV1.1 Off              # TLSv1 & TLSv1.1 are on by default
  TLSV1   Off              # TLSv1 & TLSv1.1 are on by default
  SSLV3   Off              #
  SSLV2   Off              #
  }
 }
 ##
#
 TTLSCipherParms         TLS_12
 {
  # RSA, DSS auth with AES and SHA-2
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA256
# RSA, DSS auth with AES and SHA-1
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
# RSA, DSS auth with 3DES and SHA-1
V3CipherSuites TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_RSA_WITH_3DES_EDE_CBC_SHA
}
```

Figure 18. Example: PAGENT policy setup

8. After you define the AT-TLS rules, refresh or update the policy agent for the AT-TLS rules to take effect.

   If PAGENT is started before you define the AT-TLS rules, you must refresh or update PAGENT by using the MVS MODIFY command:

   F PAGENT,REFRESH
   The **REFRESH** command triggers PAGENT to reread and process the policy files. If the **FLUSH** parameter is specified on the TcpImage or discipline configuration statement, the **REFRESH** command triggers the FLUSH processing.

   Because the FLUSH processing deletes and re-installs all policies, you must use this command only if you suspect that policies have become out of sync between the TCP/IP stack and PAGENT. Note that one consequence of triggering the FLUSH processing is that policy statistics that are being collected in the TCP/IP stack are reset.

   F PAGENT,UPDATE
   The **UPDATE** command triggers PAGENT to reread and process the policy files. This command differs slightly from the **REFRESH** command. By using the **UPDATE** command, PAGENT installs and removes only the new, changed, or deleted policies from the stack. Unchanged policies are unaffected. Therefore, it is suggested to use the **UPDATE** command in most cases.

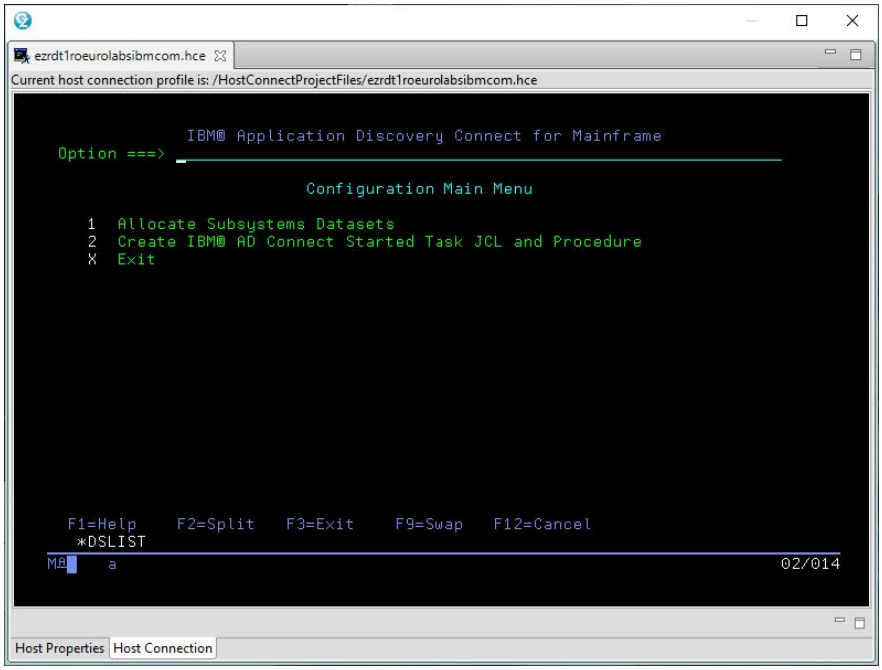
Creating Data Sets with Subsystems Datasets Allocation Panels

You can create data sets for the ChangeMan, Adabas, Natural, and Tivoli Workload Scheduler (TWS) subsystems with **Subsystems Datasets Allocation** panels. All the parameters that are used on the panels are automatically saved in your TSO user ID ISPF profile.

1. Go to the **Subsystems Datasets Allocation Main Menu** panel.
   a) Run the following command:

   ```
   TSO EX'IAYV5103.SAMP.JCL(IAYCONM0)'
   ```

   The **Configuration Main Menu** panel is displayed. You can replace `IAYV5103.SAMP.JCL` with *<site-specific-HLQ>*.SAMP.JCL. This *<site-specific-HLQ>*.SAMP.JCL library must have the IAYCONM0 member from IBM AD Connect for Mainframe.



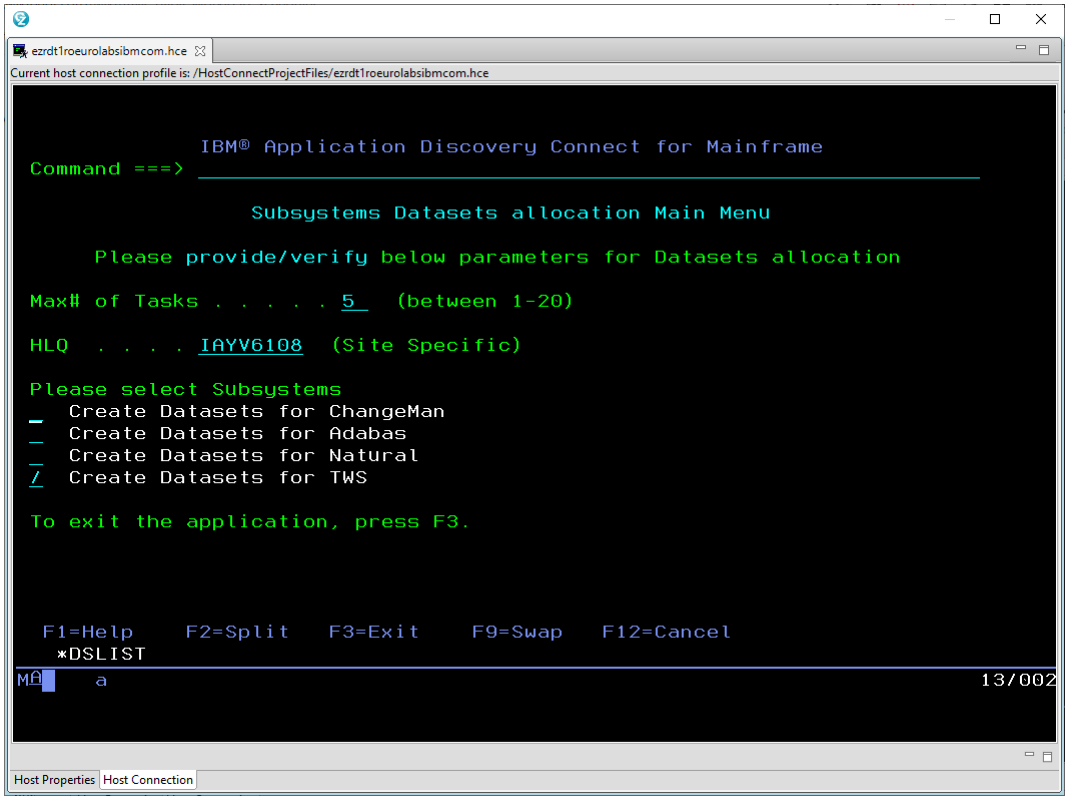
   b) On the **Configuration Main Menu** panel, select option 1.

   To end the application, select option X.

2. After the **Subsystems Datasets Allocation Main Menu** panel is opened, provide or verify the parameters for data set allocation, and then press Enter.

Notes:

- By default, the maximum number of tasks is 5. You can specify another value (1 - 20) in the **Max# of Tasks** field.
- By default, the high-level qualifier is the high-level qualifier of the **Main Menu** program. You can specify another value in the **HLQ** field.
- You must select one of the subsystems: ChangeMan, Adabas, Natural, and TWS. Otherwise, message IAYD883G will be displayed, and you cannot proceed further.

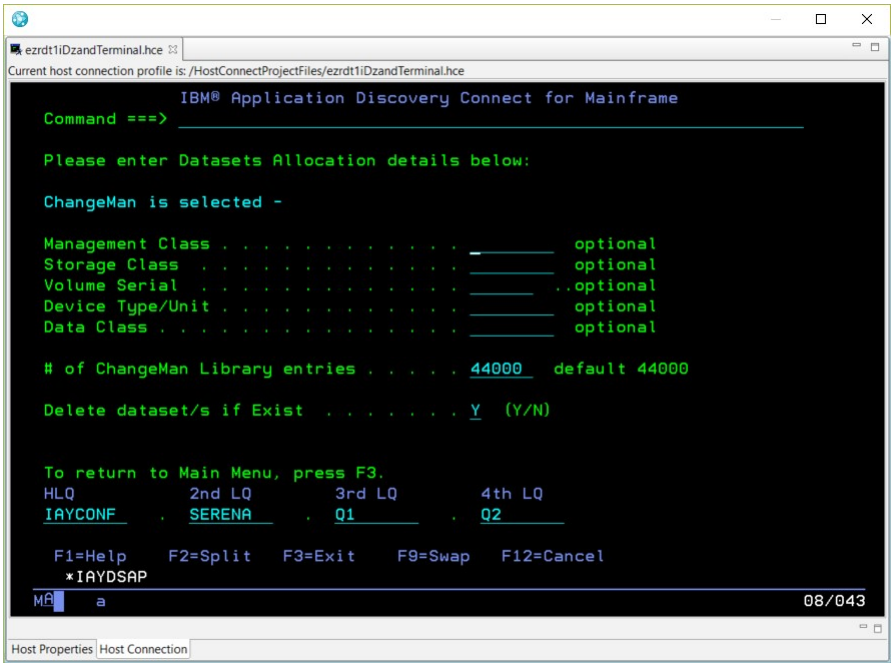


3. Provide or verify the parameters on the data set allocation panel, and then press Enter.
   - If you select Create Datasets for ChangeMan on the **Subsystems Datasets Allocation Main Menu** panel, the following panel is displayed.

     Notes:

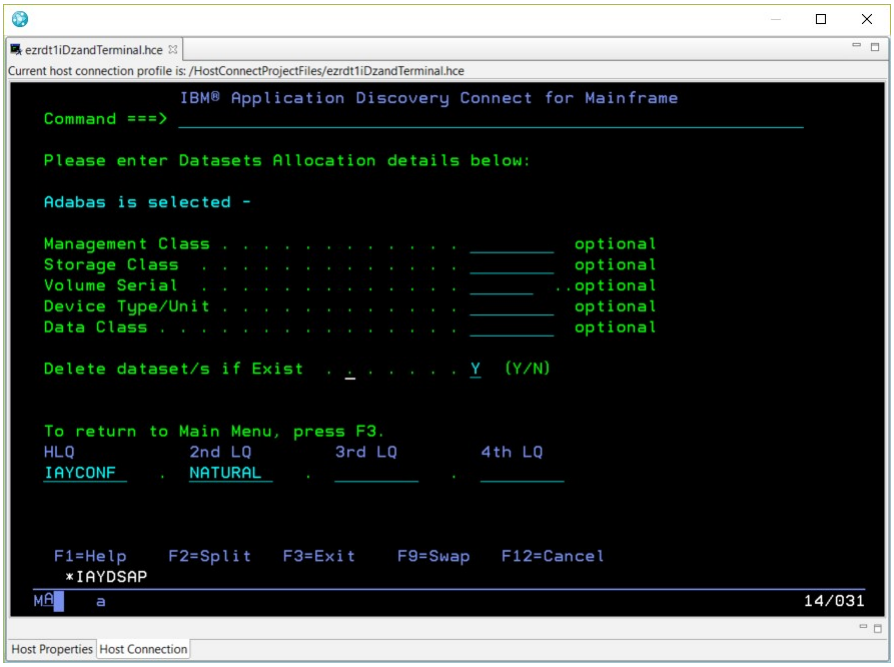
     – By default, the number of ChangeMan library entries is 44000. You can specify another value according to your site specifics in the **# of ChangeMan Library entries** field.

     – The **HLQ** and **2nd LQ** fields must be specified.



- If you select Create Datasets for Adabas on the **Subsystems Datasets Allocation Main Menu** panel, the following panel is displayed.

  Note: The **HLQ** and **2nd LQ** fields must be specified.



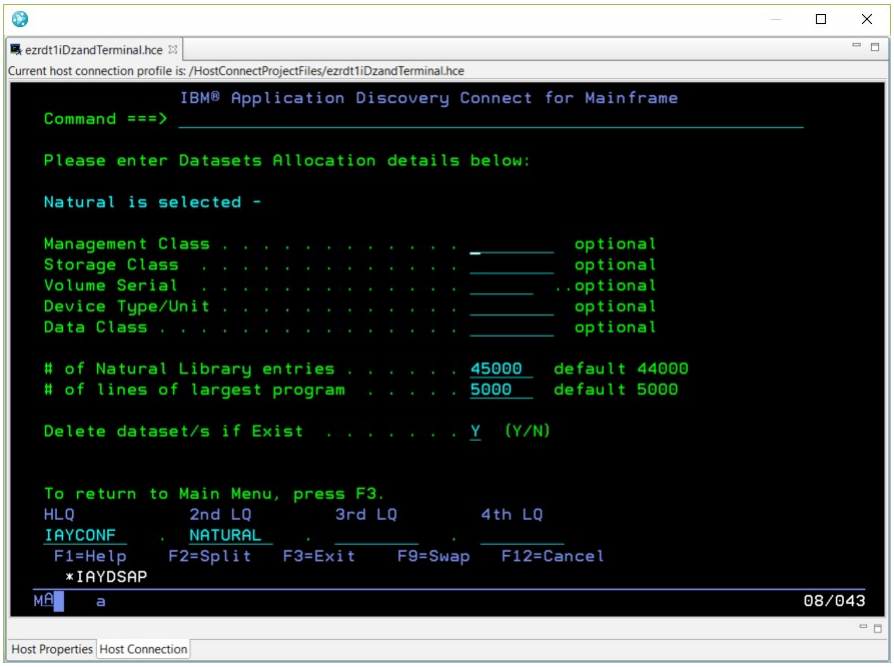
- If you select Create Datasets for Natural on the **Subsystems Datasets Allocation Main Menu** panel, the following panel is displayed.

  Notes:

  – By default, the number of Natural library entries is 44000. You can specify another value according to your site specifics in the **# of Natural Library entries** field.

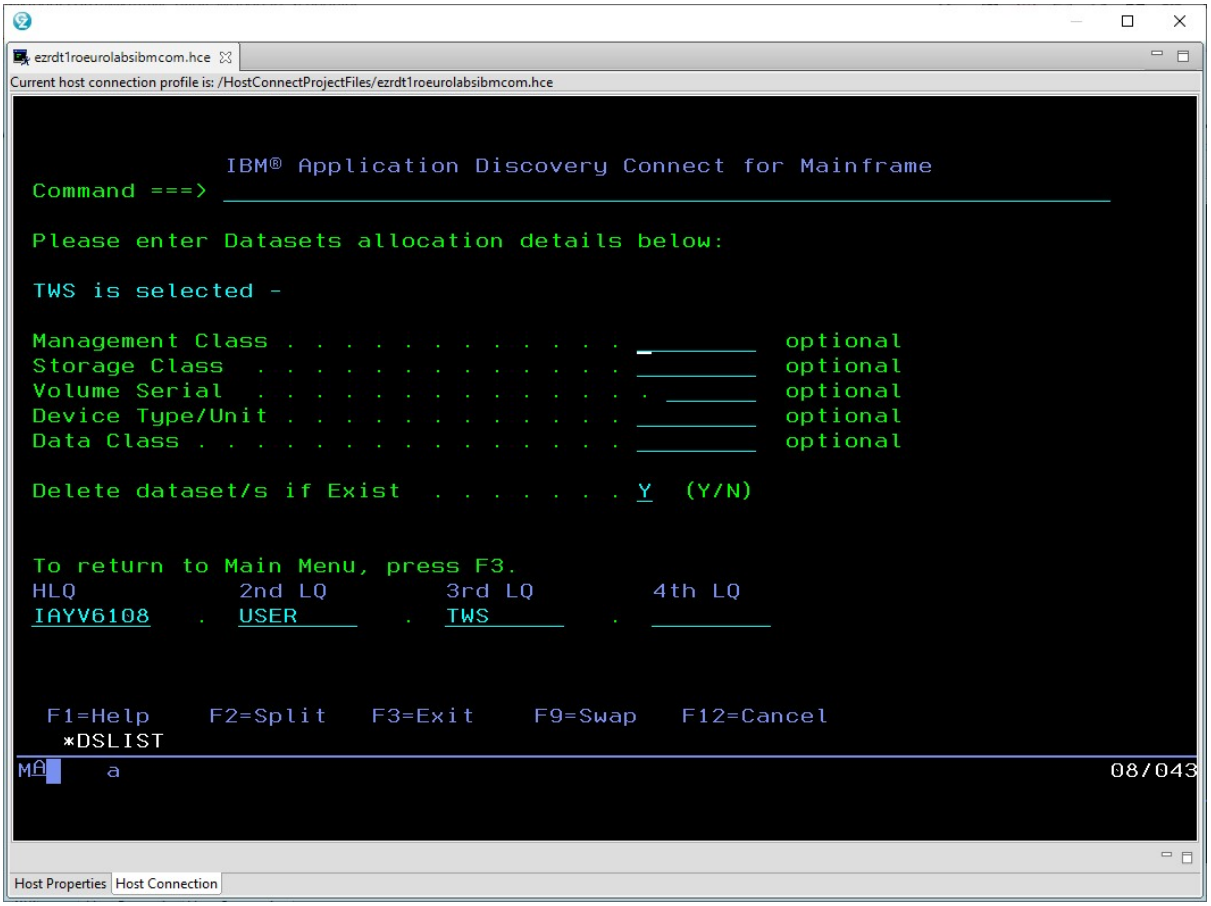
  – By default, the number of the largest program's lines is 5000. You can specify another value according to your site specifics in the **# of lines of largest program** field.

  – The **HLQ** and **2nd LQ** fields must be specified.

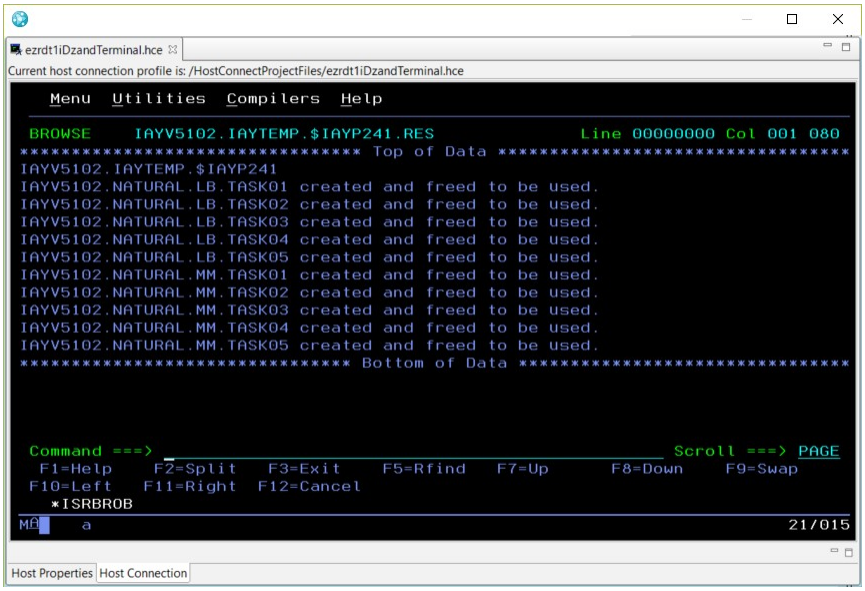


- If you select Create Datasets for TWS on the **Subsystems Datasets Allocation Main Menu** panel, the following panel is displayed.

  Note: The **HLQ**, **2nd LQ** and **3rd LQ** fields must be specified.



After data sets for the selected subsystem are created, a result panel is displayed. For example, the following image shows a result panel after data sets for the Natural subsystem are created:



To go back to the **Subsystems Datasets Allocation Main Menu** panel, press F3.

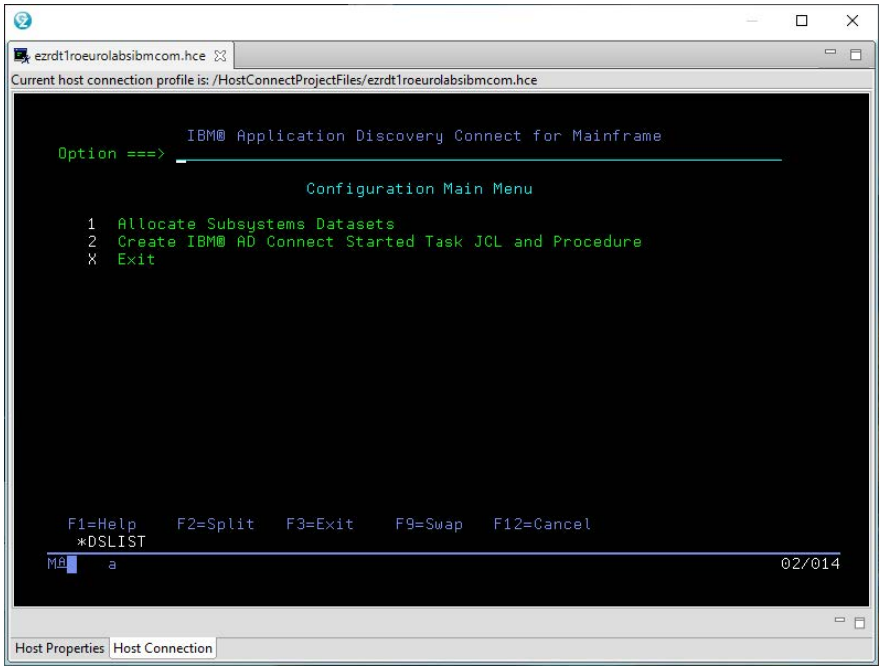
Creating Started Task JCL and Procedure Members

You can create started task JCL and procedure members with **Create IBM AD Connect Started Task JCL and Procedure** panels. All the parameters that are used on the panels are automatically saved in your TSO user ID ISPF profile.

1. Go to the **Started Task JCL/PROC creation Main Menu** panel.
   a) Run the following command:

   ```
   TSO EX'IAYV5103.SAMP.JCL(IAYCONM0)'
   ```

   The **Configuration Main Menu** panel is displayed. You can replace `IAYV5103.SAMP.JCL` with *<site-specific-HLQ>*.SAMP.JCL. This *<site-specific-HLQ>*.SAMP.JCL library must have the IAYCONM0 member from IBM AD Connect for Mainframe.

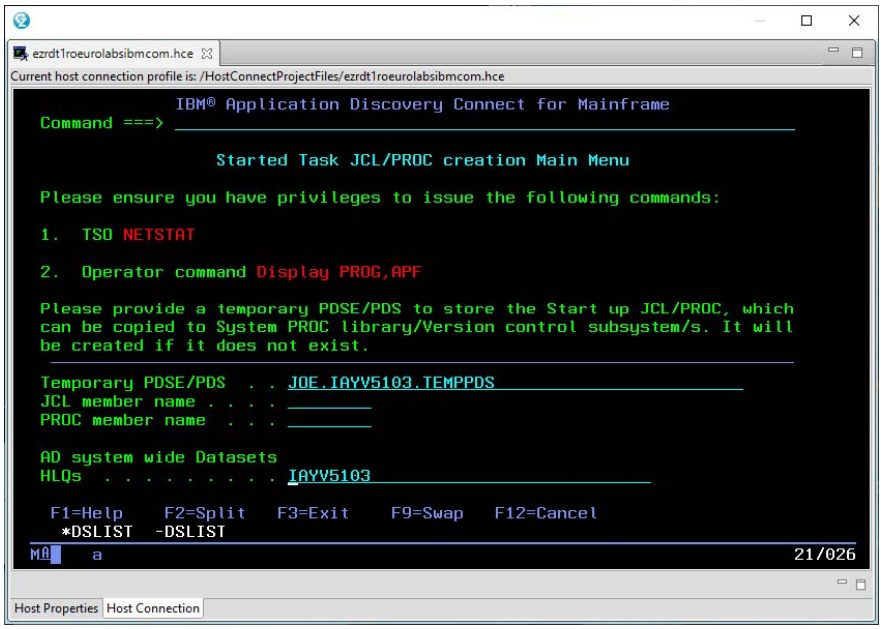


   b) On the **Configuration Main Menu** panel, select option 2.

To end the application, select option X.

2. After the **Started Task JCL/PROC creation Main Menu** panel is opened, provide or verify the parameters, and then press Enter.

   Note: You must have privileges to run commands `TSO NETSTAT` and `Display PROG,APF`.



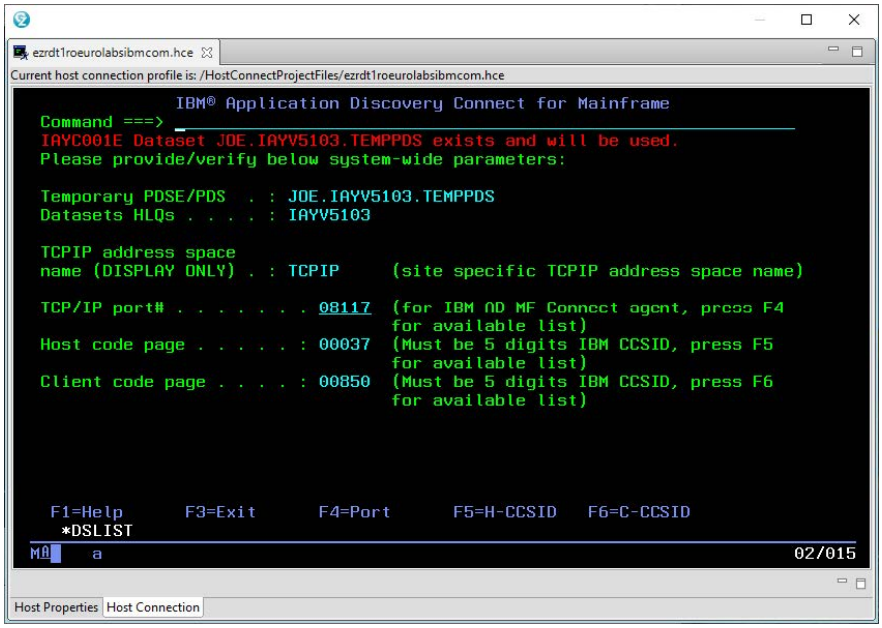
Temporary PDSE/PDS
: The name of a temporary PDSE or PDS library. By default, a value is provided in this field. You can modify the value as needed. If the specified library exists, it will be used. Otherwise, the library will be created.

JCL member name
: The name of the JCL member to be stored in the specified temporary PDSE or PDS library.

PROC member name
: The name of the procedure member to be stored in the specified temporary PDSE or PDS library.

AD system wide Datasets HLQs
: The high-level qualifiers of all the data sets in IBM AD Connect for Mainframe. By default, a value is provided in this field. You can modify the value as needed.

3. After the following panel is opened, provide or verify the parameters, and then press Enter.



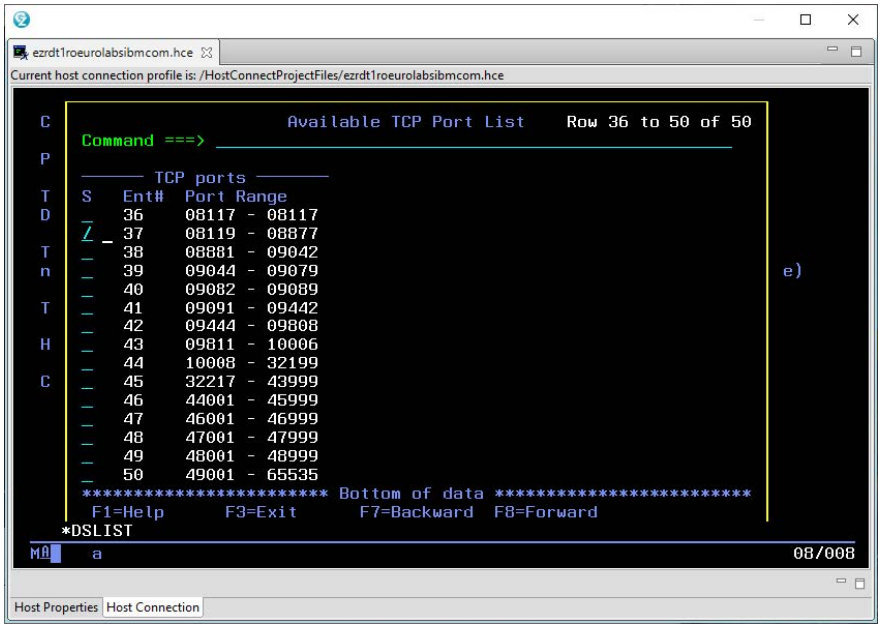
TCPIP address space name (DISPLAY ONLY)

The site-specific TCPIP address space name. This value is retrieved from the `TSO NETSTAT HOME` command response.

TCP/IP port#

The TCP/IP port number. The value must be a five-digit number, with leading zeros if required. Enter a port number directly, or select a port from the list of available ports in your system.

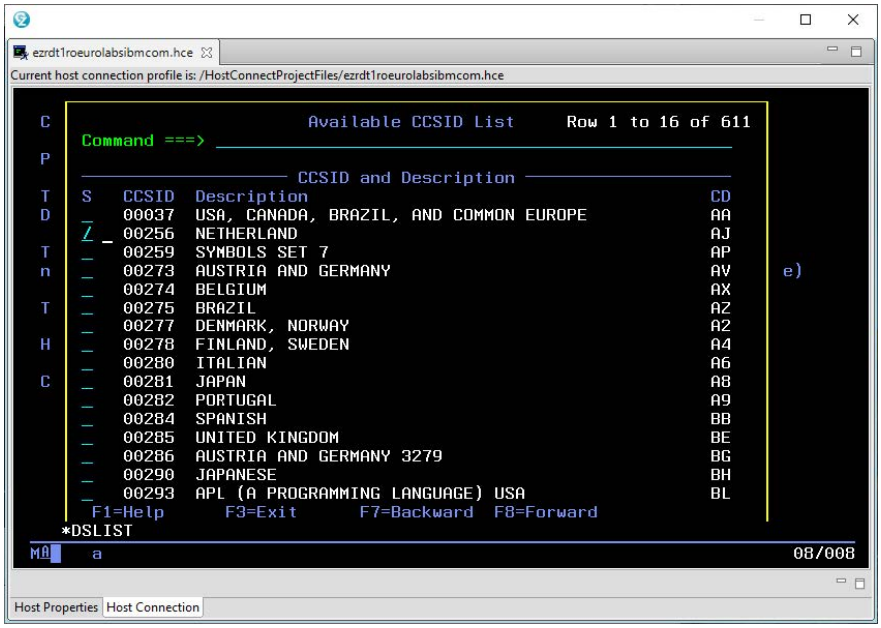
To retrieve the list of available ports, press F4. Enter a forward slash (/) in the line of the port that you want to select, and then press Enter.



Host code page

The CCSID of the host code page. Select a CCSID value from the CCSID list of available host code pages in your system.

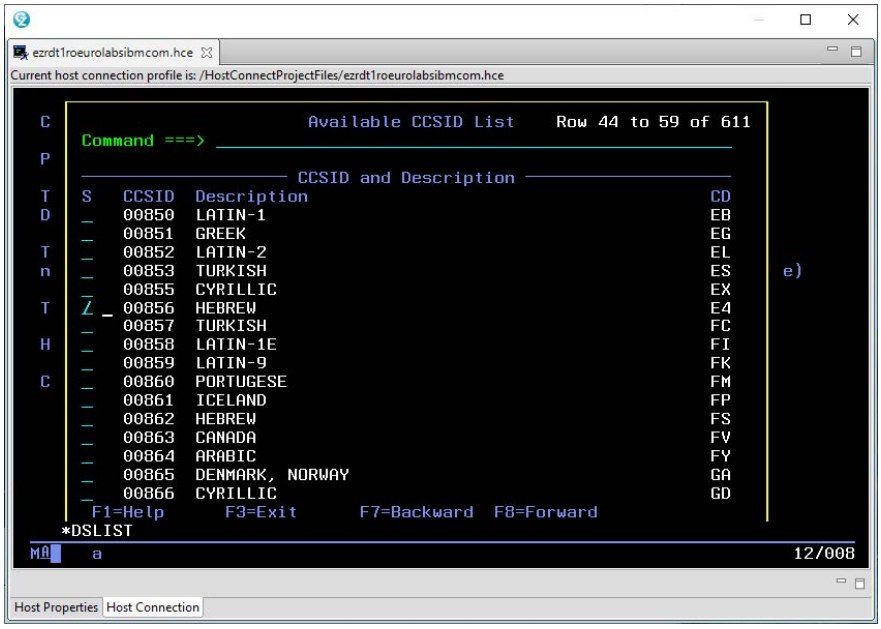
To retrieve the CCSID list of available host code pages, press F5. Enter a forward slash (/) in front of the CCSID that you want to select, and then press Enter.



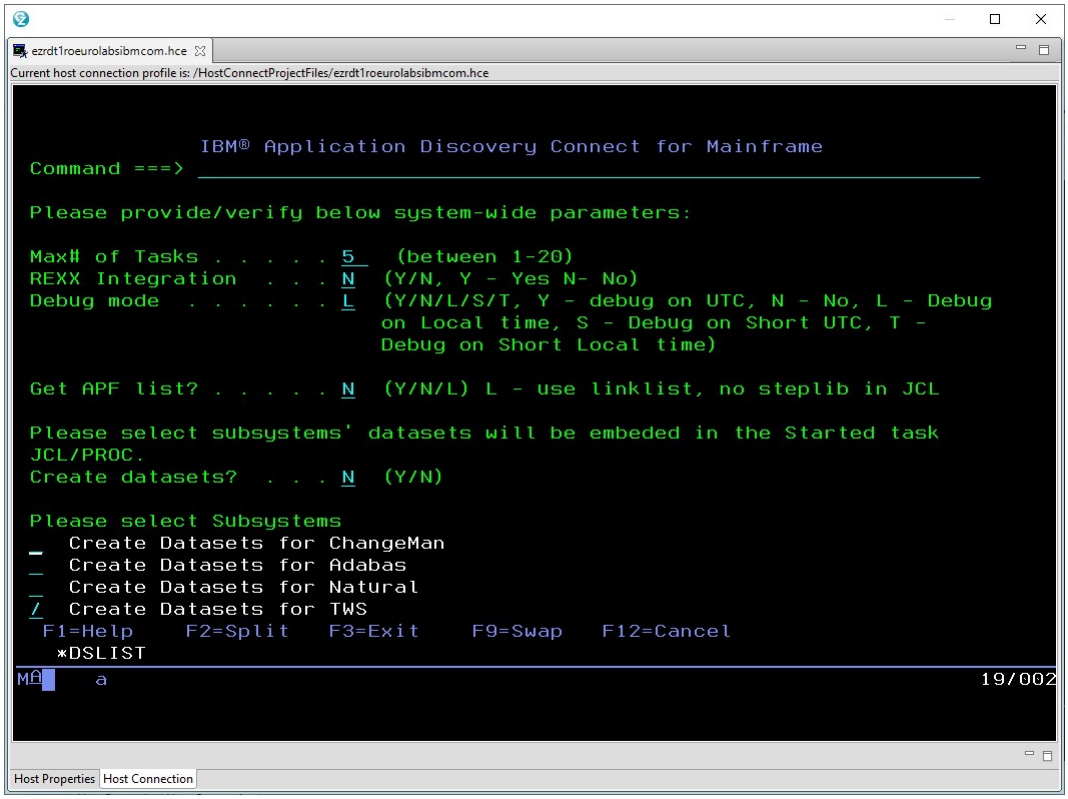
Client code page

The CCSID of the client code page. Select a CCSID value from the CCSID list of available client code pages in your system.

To retrieve the CCSID list of available client code pages, press F6. Enter a forward slash (/) in front of the CCSID that you want to select, and then press Enter.



4. After the following panel is opened, provide or verify the parameters, and then press Enter.



Max# of Tasks

The maximum number of tasks. The value must be a number in the range 1 - 20.

REXX Integration

To run CICS CSD retrieve in the batch mode, enter "Y". Otherwise, enter "N".

Debug mode

The following debug modes are available:

Value	Debug mode
Y	Debug on UTC
N	Disable debugging
L	Debug on local time
S	Debug on short UTC
T	Debug on short local time

Get APF list?

The following values are supported:

Value	Explanation
Y	Retrieve a name list of the subsystem APF libraries that will be used in the STEPLIB library of the started task.
N	Do not retrieve a name list of the subsystem APF libraries that will be used in the STEPLIB library of the started task.
L	Enter this value if the subsystem APF libraries are set up in linklist. The STEPLIB library is not required.

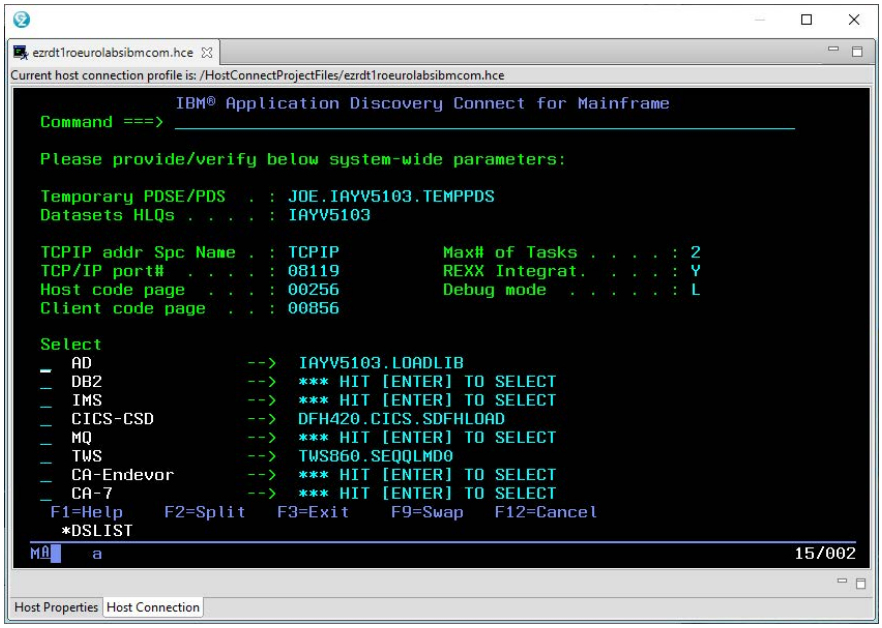
Create datasets?

To create data sets for ChangeMan, Adabas, Natural, or TWS subsystems, enter "Y". Otherwise, enter "N".

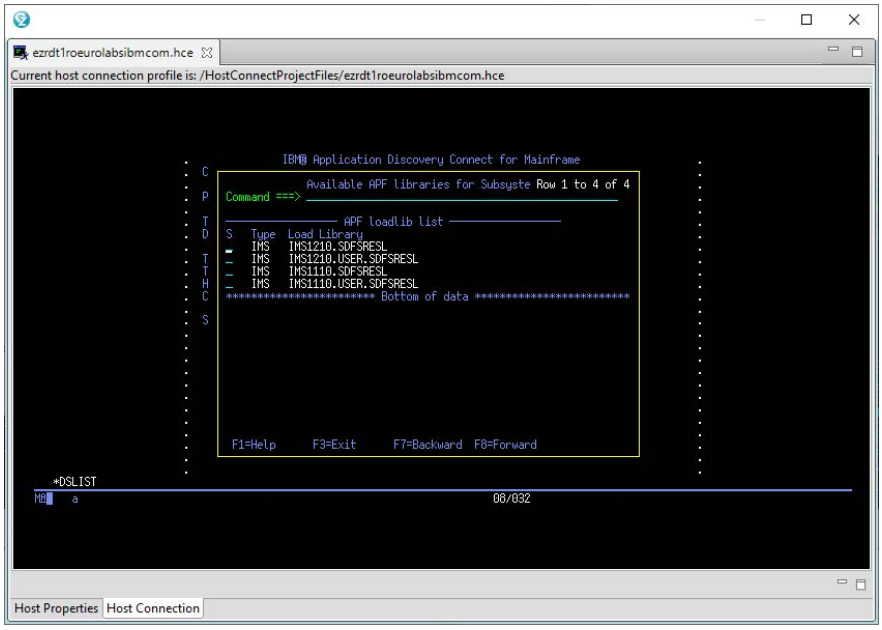
Please select Subsystems

Select one or more subsystems that you want to create data sets for by entering a forward slash (/).

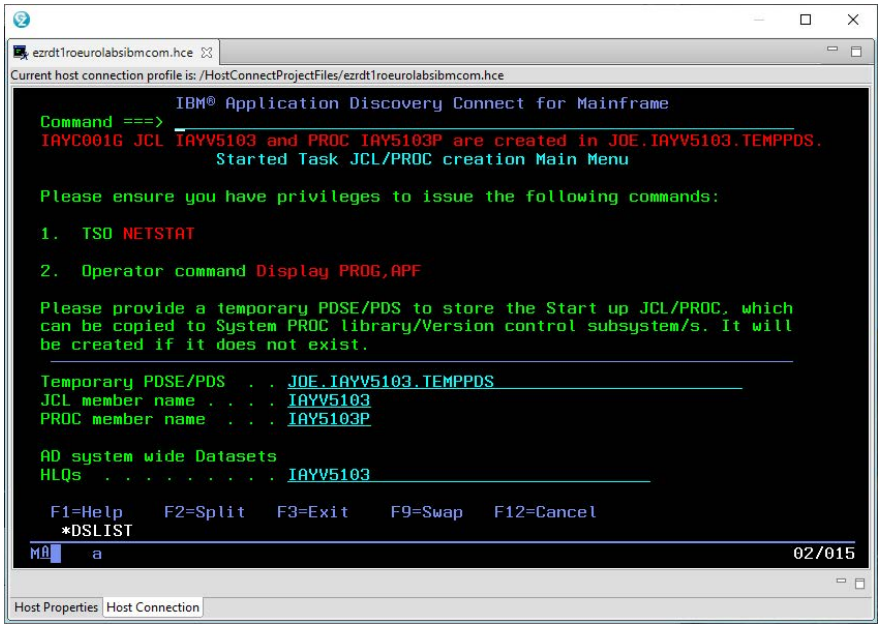
5. After the following panel is opened, select or verify the APF libraries for the subsystems. After an APF library is selected for each subsystem, press Enter.



If multiple versions of APF libraries exist in a subsystem, "*** HIT [ENTER] TO SELECT" is displayed. After you press Enter, the **Available APF libraries for Subsyste** panel is displayed. The following image shows an example of the available APF libraries for the IMS subsystem. Enter a forward slash (/) in the line of the load library that you want to select, and then press Enter. To select the first load library, you can press Enter directly.



The **Started Task JCL/PROC creation Main Menu** is displayed. Message IAYC001G, which indicates that the JCL and procedure members are created, is shown on the panel.



If "Y" is entered in the **Create datasets?** field, **Subsystems Datasets Allocation** panels are first displayed. For more information about creating data sets with **Subsystems Datasets Allocation** panels, see topic “Creating Data Sets with Subsystems Datasets Allocation Panels” on page 173.

To edit the created JCL and procedure members, use ISPF edit commands. To go back to the **Configuration Main Menu** panel, press F3.

Enabling TLS Connection to IBM AD Build

For IBM AD V5.1.0.4 and earlier versions, the communication between IBM AD Build and IBM AD Connect for Mainframe is unencrypted socket session. Beginning with version 5.1.0.5, the optional secure communication, which uses the Transport Layer Security (TLS) protocol, is supported by using the Application Transparent Transport Layer Security (AT-TLS) feature of IBM z/OS Communication Server.

The TLS protocol is a client or server cryptographic protocol. It is based on the earlier Secure Sockets Layer (SSL) specifications that are developed by Netscape Corporation for securing communications that use Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. The TLS and SSL protocols are designed to run at the application level. Therefore, typically, an application must be designed and coded to use TLS/SSL protection. On z/OS, the System SSL component of the Cryptographic Services element implements the full suite of SSL and TLS protocols (SSL V2, SSL V3, TLS V1.0, TLS V1.1, and TLS V1.2 as of this writing), including a robust set of application programming interfaces (APIs) for z/OS C and C++ applications to use.

To make the TLS or SSL protocol more accessible to z/OS applications, z/OS Communications Server V1R7 introduced the AT-TLS feature. AT-TLS starts TLS or SSL primitives in the TCP layer of the TCP/IP stack on behalf of application programs, based on policy files that describe the application traffic and how to protect it. With AT-TLS, z/OS applications that are written in almost any language can enjoy full TLS or SSL protection without requiring source code changes. AT-TLS allows socket applications to access encrypted sessions by invoking system SSL within the transport layer of the TCP/IP stack.

AT-TLS policy is read, parsed, and installed into the TCP/IP stack by the z/OS Communication Server Policy Agent (PAGENT), which implements policy-based networking for the z/OS environment. The application continues to send and receive clear text over the socket, but data that is sent over the network is protected by the System SSL component. For more information about policy-based networking, see *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

To configure IBM AD Connect for Mainframe as a server with TLS/SSL support, perform the following steps. If you have already configured AT-TLS for other servers on your z/OS system, begin at step 5.

1. Configure PAGENT as a started task on z/OS.

2. Define the security authorization for PAGENT.
3. Define PAGENT configuration files.
4. Configure AT-TLS.
5. Define the AT-TLS policy rules, and configure secure ports for IBM AD Connect for Mainframe.
6. Verify success for the PAGENT started task and policies by starting or refreshing the started task.
7. Create and configure digital certificates in RACF.
8. Provide a copy of the CA certificate that is generated in the preceding step to the person who configures IBM AD Build.

1. Configure PAGENT as a started task on z/OS.

   PAGENT runs as a UNIX process, so it can be started either from the UNIX System Service shell or as a z/OS started task. The following example shows how to start PAGENT as a z/OS started task. You can also find this sample file in TCPIP.SEZAINST(EZAPAGSP).

```
//PAGENT PROC
//PAGENT EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//STDENV DD DSN=SYS1TCPPARMS(PAGENTV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

   Figure 19. Example: PAGENT JCL started procedure

   To run with the required configuration, you can use environment variables that are configured either in an IBM Multiple Virtual Storage (IBM MVS) data set, or in a z/OS UNIX file that is specified by the STDENV data definition (DD). The following example shows how to configure the environment variables for PAGENT. You can also find this sample file in SYS1.TCPPARMS(PAGENTV).

```
TZ=PST8PDT7
PAGENT_CONFIG_FILE=//'SYS1.TCPPARMS(PAGENTP)' == > in unix default file is /etc/pagent.conf
PAGENT_LOG_FILE=SYSLOGD == > it could be /tmp/pagent.log
```

   Figure 20. Example: SYS1.TCPPARMS(PAGENTV) data set containing PAGENT environment variables

   TZ
   Specifies the local time zone for the PAGENT process. In the preceding example, it is set to Pacific Standard Time (GMT -8) and Pacific Daylight Saving Time (GMT -7).

   PAGENT_CONFIG_FILE
   Specifies the PAGENT configuration file. In the preceding example, the PAGENT configuration file is a member of the SYS1.TCPPARMS data set. For more information about PAGENT configuration files, see step 3.

   PAGENT_LOG_FILE
   Specifies the name of the log file that is used by PAGENT. In the preceding example, it is set to log PAGENT messages into file SYSLOGD.

   Before you start the PAGENT started task, make sure that the appropriate security authorizations have been set. See the following step for details.

2. Define the security authorization for PAGENT.

   The policies that are managed by PAGENT can affect system operation significantly. Therefore, you must restrict the access of z/OS user IDs to PAGENT. To do this, define certain resources and controls in the security manager product of the system, for example, RACF. To set up the security definitions of PAGENT to RACF, follow the steps:

   a. Define the PAGENT started task to RACF.

      To set up the PAGENT started task to RACF, you must define a profile to the STARTED RACF generic resource class by using the **RDEFINE** command. The following example shows the RACF commands that are used to set up the PAGENT started task. You can also find this sample file in SIAYSAMP(RACFPAG1).

```
//DAEMONS EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(STARTED)
SETROPTS RACLIST(STARTED)
SETROPTS GENERIC(STARTED)
RDEFINE STARTED PAGENT.*
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
/*
```

Figure 21. Example: RACF commands to define the PAGENT started task to RACF

If you also want to log messages into the SYSLOGD file that is mentioned in Figure 20 on page 184, use the **RDEFINE** command to define a profile for SYSLOGD to the STARTED class, for example, `RDEFINE STARTED SYSLOGD.*`.

b. Define the PAGENT user ID.

Before z/OS V1R13, the user ID for the PAGENT started task must have the z/OS UNIX superuser authority, which means the z/OS UNIX user ID (UID) for this user must be set to 0, and you must assign a default group (DFLTGRP) for the user ID. In the following example, PAGENT, the policy agent, runs under the z/OS user ID PAGENT. The RACF command is used to define the PAGENT user ID to the OMVSGRP default group with an OMVS segment that has a UID of 0. You can also find this sample file in SIAYSAMP(RACFPAG2).

```
/PAUSER EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ADDUSER PAGENT DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
/*
```

Figure 22. Example: RACF command to define a user ID for the PAGENT started task

If the SHARED.IDS profile has been defined in the UNIXPRV class, the UID value must be unique. If you want to use a UID value that is already in use, you can indicate that you intend to share the UID value across z/OS user IDs by adding the SHARED keyword to the UID parameter of the **ADDUSER** command.

If you also want to log messages into the SYSLOGD file that is mentioned in Figure 20 on page 184, use the **ADDUSER** command to define a user ID with the superuser authority for the SYSLOGD started task, for example, `ADDUSER SYSLOGD DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))`.

If you are using z/OS V1R13 or later, you can choose to use a non-superuser UID. For more information about using such a UID, see *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

c. Associate the PAGENT user ID with the PAGENT started task.

Use the RACF **RALTER** command to associate the PAGENT user ID, which is created from the preceding step, to the PAGENT started task. See the following example. You can also find this sample file in SIAYSAMP(RACFPAG3).

```
//ASCPAUSR EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RALTER STARTED PAGENT.* STDATA(USER(PAGENT))
/*
```

Figure 23. Example: RACF command to associate the PAGENT user ID with the PAGENT started task

If you also want to log messages into the SYSLOGD file that is mentioned in Figure 20 on page 184, use the **RALTER** command to associate the SYSLOGD user ID with the SYSLOGD started task, for example, `RALTER STARTED PAGENT.* STDATA(USER(SYSLOGD))`.

d. Give authorized users access to manage the PAGENT started task.

To restrict management access to the PAGENT started task, you must define a profile that is named MVS.SERVMGR.PAGENT in the OPERCMDS RACF resource class, and give only authorized users access to this facility. The following example shows the RACF commands that are used to configure this access. You can also find this sample file in SIAYSAMP(RACFPAG4).

```
//PERMITPA EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(OPERCMDS)
SETROPTS RACLIST (OPERCMDS)
RDEFINE OPERCMDS (MVS.SERVMGR.PAGENT) UACC(NONE)
PERMIT MVS.SERVMGR.PAGENT CLASS(OPERCMDS) ACCESS(CONTROL) -
ID(PAGENT)
SETROPTS RACLIST(OPERCMDS) REFRESH
/*
```

Figure 24. Example: RACF commands to give authorized users access to manage the PAGENT started task

e. Restrict access to the **pasearch** UNIX command.

The **pasearch** UNIX command can be used to display policy definitions. The output from this command indicates whether policy rules are active and shows the parsed results of the policy definition attributes. PAGENT, the policy agent is designed to ignore unknown attributes, so misspelled attributes result in the default values being used.

The **pasearch** output can be used to verify whether policies are defined correctly. However, you might not want every user to be able to see the policy definitions. To restrict unauthorized access to the **pasearch** command, a resource is defined to the SERVAUTH RACF resource class. This type of resource can be defined for each TCP/IP stack (TcpImage) and policy type (ptype) by using the following command:

```
EZB.PAGENT.sysname.TcpImage.ptype
```

sysname
The z/OS system name.

TcpImage
The procedure name of the TCP/IP stack that policy information is to be restricted to.

ptype
The policy type that is being requested. The following types are possible:

QoS
Policy Quality of Service

IDS
Policy Intrusion Detection System

IPSec
Policy Internet Protocol Security

TTLS
Policy AT-TLS

The following example shows the RACF commands that are used to restrict access to the **pasearch** UNIX command. In this example, only the USRT001 and USRT002 z/OS user IDs are permitted to use the **pasearch** command for the TCPIP TCP/IP stack on the UTEC224 z/OS system. You can also find this sample file in SIAYSAMP(RACFPAG4).

```
//PAGNACC EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RDEFINE SERVAUTH EZB.PAGENT.UTEC224.TCPIP.* UACC(NONE)
PERMIT EZB.PAGENT.UTEC224.TCPIP.* CLASS(SERVAUTH) -
ID(USRT001) ACCESS(READ)
PERMIT EZB.PAGENT.UTEC224.TCPIP.* CLASS(SERVAUTH) -
ID(USRT002) ACCESS(READ)
SETROPTS GENERIC(SERVAUTH) REFRESH
SETROPTS RACLIST(SERVAUTH) REFRESH
/*
```

*Figure 25. Example: RACF commands to restrict unauthorized users access to the **pasearch** UNIX command*

3. Define PAGENT configuration files.

   PAGENT is responsible for reading policies from configuration members in MVS data sets or in z/OS UNIX file system files. Before you can define AT-TLS policies, you must configure certain operational characteristics of PAGENT in a main configuration file. The following statements can be contained in the main configuration file:

   - The TcpImage statement
   - The LogLevel statement

   Define the TcpImage statements and appropriate logging level in the main configuration file. The following example shows definitions in the main configuration file for a single TCP/IP stack environment. You can also find this sample file in UNIX default file `/etc/pagent.conf` or in SIAYSAMP(CNFGPAG). In this example, the AT-TLS policies are defined in a separate image-specific configuration file, TTLSPOL, for the TCPIP TCP/IP image.

```
# LogLevel statement
# SYSERR, OBJERR, PROTERR, and WARNING messages are logged.
LogLevel 15
#
# TcpImage statement
# TCP/IP image: TCPIP
# Path to image-specific configuration file: SYS1.TCPPARMS(TTLSPOL)
# FLUSH parameter specified to delete existing policy data in the
# stack on PAGENT start-up or when the configuration files change.
# PURGE parameter specified to delete active policy data from the
# stack and Policy Agent when PAGENT is shut down normally.
TcpImage TCPIP //'SYS1.TCPPARMS(TTLSPOL)' FLUSH PURGE
# Here is an alternate way to specify a common AT-TLS policy for the
# TCPIP image:
#
# TcpImage TCPIP FLUSH PURGE
# TTLSConfig //'SYS1.TCPPARMS(TTLSPOL)' FLUSH PURGE
#
```

Figure 26. Example: PAGENT main configuration file

   As mentioned in Figure 20 on page 184, `/etc/pagent.conf` is copied to SYS1.TCPPARMS(PAGENTP), and this configuration file is specified by using the **PAGENT_CONFIG_FILE** environment variable in SYS1.TCPPARMS(PAGENTV).

4. Configure AT-TLS.

   AT-TLS support is enabled by specifying the TTLS parameter on the TCPCONFIG statement in the TCP/IP profile data set. The information that is required to negotiate secure connections is provided to the TCP/IP stack by the AT-TLS policies that are read, parsed, and installed by PAGENT.

   When AT-TLS is enabled and a new TCP connection is established, the AT-TLS component in the TCP layer of the stack searches for an AT-TLS rule in the policy that matches the characteristics, including local and remote IP addresses, local and remote ports, the connection direction, and other characteristics. If such a rule is found, TLS protection is applied to the rule according to the details that are specified in the AT-TLS action that is associated with the rule. If no such a rule is found, the connection is not protected with TLS.

Tip: To enable TTLS without modifying PROFILE.TCPIP, and without stopping and starting the TCP/IP stack, define a separate file that contains the TCPCONFIG TTLS statement, and run the VARY TCPIP OBEYFILE command, as shown in the following example:

```
1. OBEYFILE called TTLSON in the SYS1.TCPPARMS data set:
TCPCONFIG TTLS

2. On the MVS console, issue this command:
VARY TCPIP,,O,DSN=SYS1.TCPPARMS(TTLSON)

To disable TTLS, perform the following actions:

1. OBEYFILE called TTLSOFF in the SYS1.TCPPARMS data set:
TCPCONFIG NOTTLS

2. On the MVS console, issue this command:
VARY TCPIP,,O,DSN=SYS1.TCPPARMS(TTLSOFF)
```

To make permanent changes for TLS support, update the TCP/IP configuration in PARMLIB:

- Add the TTLS parameter to the TCPCONFIG section.
- Add the PAGENT parameter to the AUTOLOG section.

Optionally, you can set up TTLS stack initialization access control for AT-TLS. When AT-TLS is started during TCP/IP stack initialization, there might be a delay between the time that the stack comes up and when PAGENT successfully installs policy information into the stack. This situation can leave a window of time where the connections that are intended to be protected by AT-TLS can be established without that protection. While this window is open, to prevent such connections from being established, define a profile for the EZB.INITSTACK resource in the SERVAUTH RACF resource class. This type of resource can be defined for each TCP/IP stack by using the following command:

```
EZB.INITSTACK.sysname.TcpImage
```

sysname
The system name that is assigned to the z/OS LPAR.

TcpImage
The procedure name of the TCP/IP stack to which access is to be controlled.

The following example shows how to set up TCP/IP stack initialization access control for AT-TLS in the profile. You can also find this sample file in SIAYSAMP(STKINIT).

```
//STACKACC EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(SERVAUTH)
SETROPTS RACLIST (SERVAUTH)
SETROPTS GENERIC (SERVAUTH)
RDEFINE SERVAUTH EZB.INITSTACK.UTEC224.TCPIP UACC(NONE)
PERMIT EZB.INITSTACK.UTEC224.TCPIP CLASS(SERVAUTH) -
ID(OMVSKERN) ACCESS(READ)
PERMIT EZB.INITSTACK.UTEC224.TCPIP CLASS(SERVAUTH) -
ID(PAGENT) ACCESS(READ)
PERMIT EZB.INITSTACK.UTEC224.TCPIP CLASS(SERVAUTH) -
ID(SYSLOGD) ACCESS(READ)
SETROPTS GENERIC(SERVAUTH) REFRESH
SETROPTS RACLIST(SERVAUTH) REFRESH
/*
```

Figure 27. Example: RACF commands to set up TCP/IP stack initialization access control for AT-TLS

With such a profile in place, you can control which applications are allowed to establish TCP connections before PAGENT is started. To do so, provide the READ access to the EZB.INITSTACK.*sysname*.*TcpImage* profile in the SERVAUTH class for the z/OS user IDs under which such applications will run. Before connections are enabled, all other applications are forced to wait until PAGENT is initialized and its policies are installed into the stack. For more information, see *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

5. Define the AT-TLS policy rules, and configure secure ports for IBM AD Connect for Mainframe.

   `/usr/lpp/tcpip/samples/pagent_TTLS.conf` is a sample file that is provided on z/OS for this configuration. You can copy this sample file to SYS1.TCPPARMS(TTLSCONF), and then modify SYS1.TCPPARMS(TTLSCONF) to add your RACF key ring. The following example shows how to configure SYS1.TCPPARMS(TTLSCONF) for IBM AD Connect for Mainframe. Two critical configuration items in this example are the port number 8116 and the RACF key ring ADRing1. You can also find this sample file in SIAYSAMP(TTLSCONF).

   Note: The applications that are taking advantage of AT-TLS can be separated into three different types: basic, aware, and controlling. An application's type is based on whether an awareness of the service is needed and, if so, the amount of control that the application is given over the security functions. Basic applications are unchanged.

   IBM AD Connect for Mainframe is based on the basic application model. Hence, make sure that the policy for IBM AD Connect for Mainframe has the **`ApplicationControlled`** parameter set to off.

```
TTLSRule                        AD_CONNECT
{
 LocalPortRange           8116
 Direction                Inbound
 TTLSGroupActionRef       grp_Enabled
 TTLSEnvironmentActionRef  AD_Connect_Server
}
##
TTLSGroupAction                 grp_Enabled
{
 TTLSEnabled               On
 Trace                     255   # Errors to syslogd & IP joblog
#Trace                     254   # Everything to syslogd
}
##----------------------------
TTLSEnvironmentAction           AD_Connect_Server
{
 HandshakeRole Server
 TTLSKeyRingParms
 {
  Keyring ADRing1        # Keyring must be owned by the target STC ID
 }
 TTLSCipherParmsRef      TLS_12 # Ciphers modern acceptable
##
 TTLSEnvironmentAdvancedParms
 {
  ApplicationControlled Off
## TSLV1.2 only on z/OS 2.1
  TLSV1.2 On               # latest tls only
  TLSV1.1 Off              # TLSv1 & TLSv1.1 are on by default
  TLSV1   Off              # TLSv1 & TLSv1.1 are on by default
  SSLV3   Off              #
  SSLV2   Off              #
  }
 }
 ##
#
 TTLSCipherParms         TLS_12
 {
  # RSA, DSS auth with AES and SHA-2
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA256
# RSA, DSS auth with AES and SHA-1
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
```

Figure 28. Example: SYS1.TCPPARMS(TTLSCONF) configured for IBM AD Connect for Mainframe

The TTLSEnvironmentAction statement also refers to the TTLSCipherParms statement that lists the cipher suites to be allowed. The TTLSCipherParms statement specifies the list of acceptable TLS/SSL ciphers suites in the order of preference, top to bottom, most preferred to least preferred. If you do not specify a TTLSCipherParms statement, AT-TLS uses the default cipher list of the System SSL component.

Notes:

- The default cipher list contains some cipher suites that are considered to be rather weak by the current cryptographic standards, for example, single Data Encryption Standard (DES), MD5, and RC2. So it's suggested to specify a TTLSCipherParms statement that enforces a minimum cryptographic strength that meets your company's security policies.

- If you need to comply with the US National Institute of Standards and Technology (NIST) Special Publication SP800-131a, specify a TTLSCipherParms statement that includes only compliant cipher suites. The cipher suite that is shown in Figure 29 on page 193 complies with NIST SP800-131a, with preference given to the strongest suites.
- If you need to comply with NIST Special Publication SP800-52, as part of the compliance with Federal Information Processing Standards Security Requirements for Cryptographic Modules, which is referred to as FIPS 140-2, change the TTLSGRoupAction statement from `FIPS140 Off` to `FIPS140 On`. This change ensures that when Db2 for z/OS is acting as a client, only ciphers that are compliant with the FIPS140-2 requirements are allowed.
- Another reason why you might need to specify the list of cipher suites is to match the capabilities of your communications partners. For a TLS/SSL handshake to succeed, the server and client must support at least one cipher suite in common, so you might need to customize your server's cipher suite list to include a cipher that your client supports.

For more information about other AT-TLS policy statements and AT-TLS policy configurations, see section "Application Transparent Transport Layer Security data protection" in *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*. For detailed syntax of each AT-TLS policy statement, see the topics about AT-TLS policy statements, the policy agent, and policy applications in *z/OS Communications Server V2R3: IP Configuration Guide, SC27-3650-30*.

6. Verify success for the PAGENT started task and policies by starting or refreshing the started task.

   To start the PAGENT started task, run the `S PAGENT` command. To stop the PAGENT started task, run the `P PAGENT` command. If the PURGE option is specified in the policy configuration files, when PAGNET is stopped, all policies from the TCP/IP stack are deleted.

   If PAGENT is started before you define the AT-TLS rules, you must refresh or update PAGENT by using the MVS MODIFY command:

   F PAGENT,REFRESH
   The **REFRESH** command triggers PAGENT to reread and process the policy files. If the **FLUSH** parameter is specified on the TcpImage or discipline configuration statement, the **REFRESH** command triggers the FLUSH processing.

   Because the FLUSH processing deletes and re-installs all policies, you must use this command only if you suspect that policies have become out of sync between the TCP/IP stack and PAGENT. Note that one consequence of triggering the FLUSH processing is that policy statistics that are being collected in the TCP/IP stack are reset.

   F PAGENT,UPDATE
   The **UPDATE** command triggers PAGENT to reread and process the policy files. This command differs slightly from the **REFRESH** command. By using the **UPDATE** command, PAGENT installs and removes only the new, changed, or deleted policies from the stack. Unchanged policies are unaffected. Therefore, it is suggested to use the **UPDATE** command in most cases.

   To query information from PAGENT, use the **pasearch** z/OS UNIX command. This command can be issued from the UNIX Systems Services shell or from the Time Sharing Option Extensions (TSO/E) **oshell** command. The following example shows the output from issuing the **oshell** command to run the **pasearch** command that specifies the `-t` option to display all AT-TLS policy entries:

```
policyRule:             AD_CONNECT
   Rule Type:            TTLS
   Version:              3                 Status:            Active
   Weight:               1                 ForLoadDist:       False
   Priority:             1                 Sequence Actions:  Don't Care
   No. Policy Action:    2
   policyAction:         grp_Enabled
    ActionType:          TTLS Group
    Action Sequence:     0
   policyAction:         AD_Connect_Server
    ActionType:          TTLS Environment
    Action Sequence:     0
   Time Periods:
    Day of Month Mask:
    First to Last:       11111111111111111111111111111111
    Last to First:       11111111111111111111111111111111
    Month of Yr Mask:    111111111111
```

```
    Day of Week Mask:    1111111  (Sunday - Saturday)
    Start Date Time:     None
    End Date Time:       None
    Fr TimeOfDay:        00:00             To TimeOfDay:      24:00
    Fr TimeOfDay UTC:    00:00             To TimeOfDay UTC:  00:00
    TimeZone:            Local
TTLS Condition Summary:                 NegativeIndicator: Off
 Local Address:
  FromAddr:           All
  ToAddr:             All
 Remote Address:
  FromAddr:           All
  ToAddr:             All
 LocalPortFrom:       8116              LocalPortTo:       8116
 RemotePortFrom:      0                 RemotePortTo:      0
 JobName:                               UserId:
 ServiceDirection:    Inbound
Policy created: Wed Oct 30 10:52:29 2019
Policy updated: Wed Oct 30 10:52:29 2019

TTLS Action:                   grp_Enabled
  Version:                       3
  Status:                        Active
  Scope:                       Group
  TTLSEnabled:                 On
  CtraceClearText:              Off
  Trace:                         255
  FIPS140:                      Off
  TTLSGroupAdvancedParms:
      SecondaryMap:               Off
      SyslogFacility:             Daemon
     Policy created: Wed Oct 30 10:52:29 2019
     Policy updated: Wed Oct 30 10:52:29 2019

   TTLS Action:                   AD_Connect_Server
     Version:                     3
     Status:                      Active
     Scope:                       Environment
     HandshakeRole:               Server
     SuiteBProfile:               Off
     TTLSKeyringParms:
      Keyring:                    ADRing1
     TTLSEnvironmentAdvancedParms:
      SSLv2:                      Off
      SSLv3:                      Off
      TLSv1:                      Off
      TLSv1.1:                    Off
      TLSv1.2:                    On
      ApplicationControlled:      Off
      HandshakeTimeout:           10
      ClientAuthType:             Required
      ResetCipherTimer:           0
      TruncatedHMAC:              Off
      CertValidationMode:         Any
      ServerMaxSSLFragment:       Off
      ClientMaxSSLFragment:       Off
      ServerHandshakeSNI:         Off
      ClientHandshakeSNI:         Off
      Renegotiation:              Default
      RenegotiationIndicator:     Optional
      RenegotiationCertCheck:     Off
      3DesKeyCheck:               Off
      ClientEDHGroupSize:         Legacy
      ServerEDHGroupSize:         Legacy
      PeerMinCertVersion:         Any
      PeerMinDHKeySize:           1024
      PeerMinDsaKeySize:          1024
      PeerMinECCKeySize:          192
      PeerMinRsaKeySize:          1024
      ServerScsv:                 Off
     TTLSCipherParms:
      v3CipherSuites:
       C02B  TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
       C02F  TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
       C00A  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA


      C009  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
      C013  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
      C014  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
      0039  TLS_DHE_RSA_WITH_AES_256_CBC_SHA
      0035  TLS_RSA_WITH_AES_256_CBC_SHA
```

```
      C02C  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
      C024  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
      C023  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
      C030  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
      C028  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
      C027  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
    TTLSGskAdvancedParms:
     TTLSGskHttpCdpParms:
      HttpCdpEnable:            Off
      HttpCdpProxyServerPort:   80
      HttpCdpResponseTimeout:   15
      HttpCdpMaxResponseSize:   204800
      HttpCdpCacheSize:         32
      HttpCdpCacheEntryMaxsize: 0
     TTLSGskOcspParms:
      OcspAiaEnable:            Off
      OcspProxyServerPort:      80
      OcspRetrieveViaGet:
Off
 ***
```

Figure 29. Example: displaying AT-TLS policies by using the `pasearch -t` command

If the configurations are successful, no glaring errors occur in the PAGENT started task output, and no errors occur in the `/tmp/pagent.log` or SYSLOGD log file. The log file is set up in SYS1.TCPPARMS(PAGENTV) as shown in Figure 20 on page 184.

Several tools can be used to diagnose AT-TLS problems. For a complete description of common AT-TLS start-up errors, debugging steps and tools, and the descriptions of AT-TLS return codes, see the Diagnosing Application Transparent Transport Layer Security (AT-TLS) section in *z/OS Communications Server V2R3: IP Diagnosis Guide, GC27-3652-30*.

7. Create and configure digital certificates in RACF.

   You can create and manage digital certificates and their related key pairs in many ways. On z/OS, the following options are provided:

   - z/OS Security Server RACF and other SAF-compliant external security managers
   - The gskkyman tool that is included with the System SSL component
   - z/OS Cryptographic Services Public Key Infrastructure (PKI) Services

   gskkyman and RACF are suited for smaller scale certificate deployments, while PKI Services is a full-blown certificate authority (CA) and management system that can manage digital certificates on an enterprise level. In this document, RACF is used to generate and manage certificates and keys in SAF key rings on z/OS.

   Digital certificates and key rings are managed in RACF primarily by using the **RACDCERT** command. The following example shows how to use this command to create a key ring, a CA certificate that is connected to the key ring, and a personal certificate that is signed with the CA certificate. The personal certificate is also connected to the key ring. You can also find this sample file in SIAYSAMP(DCERTJCL).

   CAUTION: Use this sample file with caution because this process deletes objects. Adjust the JCL as necessary to fit your system's configuration.

```
//MAKECERT JOB 'SAMPLE JCL', MSGLEVEL=1,MSGCLASS=H,CLASS=A,REGION=0M
//*****************************************************************
//* WARNING! This job deletes rings and certificates!!! *
//* ------------------------------------------------------------- *
//* Job instructions: *
//* 1. Enter valid jobcard above. *
//* 2. Change ++MY_KEYRING++ to your key ring name. Ex: ADRing1 *
//* 3. Change ++MY_CA_CERT++ to your CA cert name. Ex: SystemRootCA *
//* 4. Change ++MY_CERT++ to your non-CA cert. Ex: ADAgentPersonalCert*
//* 5. Change ++HLQ++ to your personal HLQ on system. *
//* 6. Change ++NOTBEFORE++ to a not-before date YYYY-MM-DD. *
//* 7. Change ++NOTAFTER++ to a not-after date YYYY-MM-DD. *
//* 8. Submit. You should receive RC=0 for every step. *
//* ------------------------------------------------------------- *
//* At the end of this job you should have the following: *
//* Keyring : ++MY_KEYRING++ *
//* CA Cert : ++MY_CA_CERT++ (Connected to ring) *
//* Non-CA Cert: ++MY_CERT++ (Connected to ring) *
```

```
//*********************************************************************
//*********************************************************************
//* Delete CA Cert if it exists. *
//*********************************************************************
//STEP010 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT CERTAUTH DELETE(LABEL('++MY_CA_CERT++')) ID(STCSYS)
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
/*
//*********************************************************************
//* Create CA Cert. *
//*********************************************************************
//STEP020 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT CERTAUTH GENCERT -
SUBJECTSDN( o('IBM Corporation') -
ou('Z Certificate Authority') -
C('US')) -
NOTBEFORE(DATE(++NOTBEFORE++)) -
NOTAFTER(DATE(++NOTAFTER++)) -
KEYUSAGE(CERTSIGN) -
WITHLABEL('++MY_CA_CERT++')
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
/*
//*********************************************************************
//* Delete a Personal Cert. *
//*********************************************************************
//STEP030 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT ID(STCSYS) DELETE(LABEL('++MY_CERT++')
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
/*
//*********************************************************************
//* Create a Personal certificate and reference the CA cert above. *
//*********************************************************************
//STEP040 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT GENCERT ID(STCSYS) +
SUBJECTSDN(CN('AD Connect') +
O('International Business Machines Corporation') C('US')) +
SIZE(2048) NOTBEFORE(DATE(++NOTBEFORE++) TIME(11:00:00)) +
NOTAFTER(DATE(++NOTAFTER++) TIME(11:00:00)) +
WITHLABEL('++MY_CERT++') +
RSA KEYUSAGE(CERTSIGN DATAENCRYPT HANDSHAKE) +
SIGNWITH(CERTAUTH LABEL('++MY_CA_CERT++'))
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
/*

//*********************************************************************
//* List the CA cert. *
//*********************************************************************
//STEP050 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT CERTAUTH LIST(LABEL('++MY_CA_CERT++'))
/*
//*********************************************************************
//* Delete key ring if it exists. *
//*********************************************************************
//STEP070 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT ID(STCSYS) DELRING(++MY_KEYRING++)
/*
//*********************************************************************
//* Create a key ring. *
//*********************************************************************
//STEP080 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT ID(STCSYS) ADDRING(++MY_KEYRING++)
/*
//*********************************************************************
```

```
//* List the key ring. *
//*********************************************************************
//STEP090 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT LISTRING(++MY_KEYRING++) ID(STCSYS)
/*
//*********************************************************************
//* CONNECT THE Personal Cert TO THE RING. *
//*********************************************************************
//STEP100 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT ID(STCSYS) CONNECT(LABEL('++MY_CERT++') -
RING(++MY_KEYRING++) DEFAULT)
/*
//*********************************************************************
//* CONNECT THE CA Cert to the ring. *
//*********************************************************************
//STEP110 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT ID(STCSYS) CONNECT(CERTAUTH -
LABEL('++MY_CA_CERT++') -
RING(++MY_KEYRING++) -
USAGE(CERTAUTH))
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
/*
//*********************************************************************
//* Display the key ring which should now have 2 certs connected. *
//*********************************************************************
//STEP120 EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT LISTRING(++MY_KEYRING++) ID(STCSYS)
/*
//*********************************************************************
//*********************************************************************
//* At this point, you should have your key ring and certs connected. *
//* If you don't something went wrong. Start over. Don't proceed. *
//*********************************************************************
//*********************************************************************
//*********************************************************************
//* Export CA to a file so it can be sent to your PC and save it in
//* AD CONNECT Bin Folder.
//* It can be exported using RACF ISPF panels as well.
//*********************************************************************
//STEP130 EXEC PGM=IKJEFT01,DYNAMNBR=50
//SYSLBC DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RACDCERT CERTAUTH EXPORT(LABEL('++MY_CA_CERT++')) -
FORMAT(CERTB64) DSN('++HLQ++.CACERT.EXPORT') -
/*
```

Figure 30. Example: sample JCL to create a key ring, a CA certificate, and a personal certificate

Important: When creating the personal certificate that will be presented to the client, which is IBM AD Build Client in this case, on behalf of the IBM AD Connect for Mainframe started task, set the Subject Name CN field equal to the numeric IP address of the z/OS mainframe instance where the task is running. Do not use the host name. For example, use >CN=9.30.123.456.OU=IBM CA.C=US< instead of >CN=zos.server.name.com.OU=IBM CA.C=US<.

For a complete description of the **RACDCERT** command and all of the functions that this command supports, see the "RACF command syntax" section in *z/OS V2R3: Security Server RACF Command Language Reference, SA23-2292-30*.

8. Provide a copy of the CA certificate that is generated in the preceding step to the person who configures IBM AD Build.

   Ensure that you preserve the selected format when the CA certificate is generated. In the preceding example, the format is CERTB64. The CERTB64 format creates a DER encoded X.509 certificate that is in the Base64 format, so download this certificate in the ASCII mode if using FTP. An alternative to FTP

is to cut and paste the certificate contents from the mainframe editor screen (ISPF) to a PC-based text editor, and then save the resulting file.

During the TLS handshake process, the copy of the CA certificate will be used by IBM AD Build to authenticate the personal certificate that is presented on behalf of IBM AD Connect for Mainframe.

IBM AD Build can establish a secure communication with the mainframe host by using the OpenSSH toolkit. To enable TLS connection between IBM AD Build and IBM AD Connection for Mainframe after you complete the AT-TLS setup for IBM AD Connect for Mainframe, the configurations for IBM AD Build must also be completed. For instructions, see "Enabling TLS Connection between IBM AD Build Client and IBM AD Connect for Mainframe" on page 291.

Enabling the z/OS User Authentication feature for IBM AD Connect for Mainframe

Introduction

Starting with **IBM AD V6.0.1**, **IBM AD Build Client**, **IBM AD Build Configuration** and **IBM Connect for Mainframe** can be configured to require a user to provide a valid z/OS user ID and password before you access the project resources on the mainframe.

When configured for user authentication, **IBM AD Build Client** will prompt the user to provide a valid z/OS user ID and password. **IBM AD Build Client** will then include these credentials in all subsequent requests that are sent to the mainframe. **IBM AD Connect for Mainframe** will extract these credentials and perform authentication by using the installed z/OS security product (RACF, ACF2, or others).

If the authentication is successful, the user's requests will be serviced by a subtask running under the security context of the authenticated z/OS user ID. The authenticated user ID must have the appropriate access privileges in place to retrieve the requested resources.

If the authentication fails, an error message is sent back to the **IBM AD Build Client** user. No attempt will be made to access the requested resources.

In the previous releases, all mainframe requests were serviced by using the STC ID of the AD Connect for Mainframe started task.

Current limitations

For the **IBM AD V6.0.1** release, only a standard of 8 character mixed case password is supported. Password phrases are not supported.

The user ID and password can only be input through the **IBM AD Build Client** and **IBM AD Build Configuration** User Interfaces (GUI). Setting these fields in the command-line interface (CLI) is not currently supported.

Compatibility with earlier versions of IBM AD

IBM AD Build Client version 6.0.1 will not function with earlier versions of **IBM AD Connect for Mainframe**.

IBM AD Connect for Mainframe version 6.0.1 will not function with earlier versions of **IBM AD Build Client**.

Note: If you want to use IBM AD V6.0.1, make sure that you have installed and configured the latest version of **IBM AD Build Client** and **IBM AD Connect for Mainframe**.

Recommendations for the z/OS User Authentication feature

Before enabling the **z/OS User Authentication** feature, you need to make sure that the AT-TLS setup for IBM AD Connect for Mainframe is **completed** and the TLS connection between **IBM AD Build Client** and **IBM AD Connect for Mainframe** is **enabled**. **IBM AD Build Client** will not send the z/OS user credentials over an unencrypted TCP/IP connection. For more information, see "Enabling TLS Connection to IBM AD

Build” on page 183 and “Enabling TLS Connection between IBM AD Build Client and IBM AD Connect for Mainframe” on page 291.

If you want to use the **z/OS User Authentication** feature, it must be enabled on both **IBM AD Build Client** and **IBM AD Build Configuration** as well as for **IBM AD Connect for Mainframe** on z/OS. For more information, see “Enabling the z/OS User Authentication feature for IBM AD Build Client” on page 294 and “Enabling the z/OS User Authentication feature for IBM AD Build Configuration” on page 238.

If you do not want to use the **z/OS User Authentication** feature, it must be disabled on **IBM AD Build Client**, **IBM AD Build Configuration**, and **IBM AD Connect for Mainframe**.

Important: After the installation and configuration of **IBM AD V6.0.1** the **z/OS User Authentication** feature is **disabled** by default. There are no actions to be performed in case that this is the desired configuration.

Enabling the z/OS User Authentication feature for IBM AD Connect for Mainframe

The tenth positional parameter in the **IBM AD Connect for Mainframe** started task procedure PARM list is the switch that enables or disables the **z/OS User Authentication** feature. The default value is **N** (authentication disabled) and you must set it to **Y** (authentication enabled) to enable the z/OS User authentication feature.

Note: When the value of this PARM is changed, the **IBM AD Connect for Mainframe** started task must be restarted to read in the new value.

The following sample is available in member IAYLSTNR in the SIAYSAMP distribution library of **IBM AD Connect for Mainframe**.

```
 PARM='&TCPAS,&PORT,&MAXTK,&CONNNAME,&DEB,&HOSTCP,&CLNTCP,N,&RXI,Y'
*********************************************************************
*
* PARM      DESCRIPTION
* POSITION
* ----------------------------------------------------------------
* 1         NAME OF TCPIP STARTED TASK ON Z/OS
* 2         PORT NUMBER ON WHICH AD CONNECT FOR MAINFRAME LISTENS
* 3         MAXIMUM # OF WORKER SUBTASKS FOR AD CONNECT FOR MAINFRAME
* 4         MAIN REQUEST HANDLER PROGRAM - ALWAYS IAYCONN.
* 5         DEBUG TRACE ON/OFF SWITCH (DEFAULT N)
* 6         MAINFRAME (EBCDIC) CODE PAGE
* 7         CLIENT(WINDOWS)    CODE PAGE
* 8         RESERVED (FORMERLY DBCS SWITCH - DOUBLE BYTE CHAR)
* 9         REXX INTEGRATION SWITCH (DEFAULT N)
*10         Z/OS USERID/PASSWORD AUTHENTICATION SWITCH (DEFAULT N)
*
```

Figure 31. IAYLSTNR member

Grant user ID access to necessary mainframe resources

After a successful login to **IBM AD Connect for Mainframe**, all access to mainframe resources will be made under the security context of the logged in user. Therefore, you must ensure that this user ID has the necessary read level access to all mainframe resources that will be retrieved for the corresponding AD project.

Attention:

There is one exception. First time when **IBM AD Build Client** retrieves project information from the mainframe, it will request **IBM AD Connect for Mainframe** to issue two MVS commands.

```
DISPLAY M=CPU
DISPLAY SYMBOLS
```

These commands will run under the started task user ID of the IBM AD Connect for Mainframe task. It is not expected that the z/OS user ID of a typical AD Project administrator will have (or even should have) authority to issue MVS system commands, although they are DISPLAY commands only.

Verifying the z/OS User Authentication feature configuration

1. If the **z/OS User Authentication** feature is **enabled**, you will see the `IAYMF0050I` message in the `CPEOUT` file when IBM AD Connect for Mainframe is started.

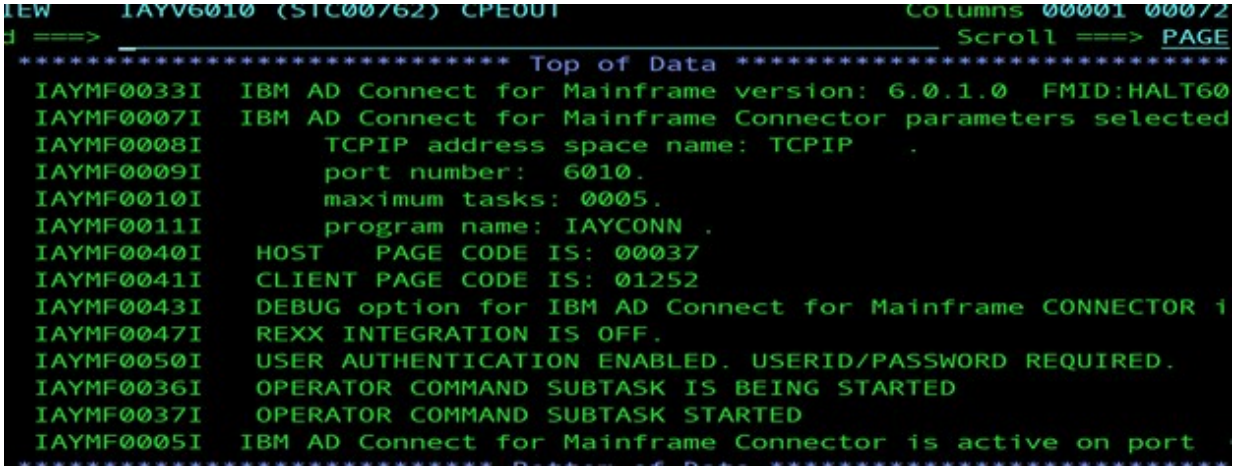


Figure 32. IAYMF0050I message

2. If the **z/OS User Authentication** feature is **disabled**, you will see the `IAYMF0051I` message in the `CPEOUT` file when IBM AD Connect for Mainframe is started.

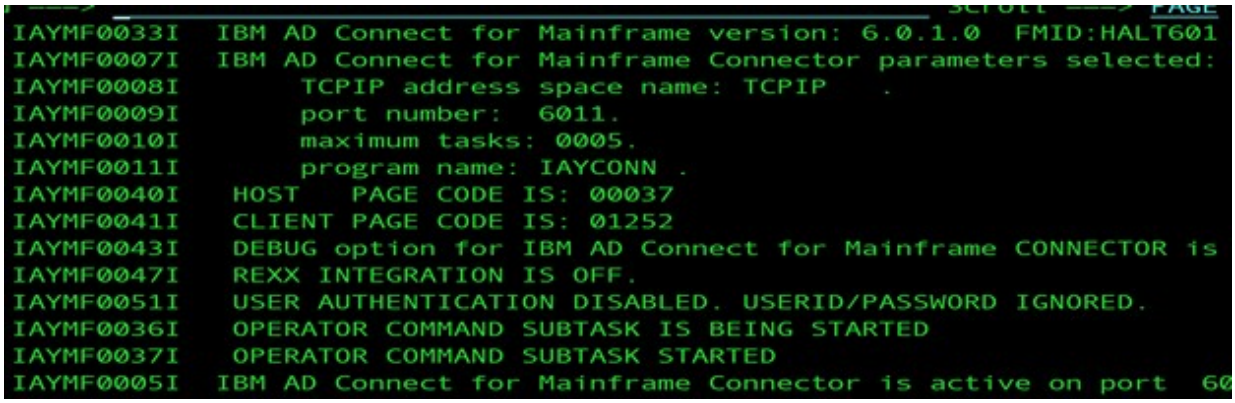


Figure 33. IAYMF0051I message

As previously stated, ensure that **IBM AD Build Client** is also configured to match the authentication setting (Y/N) used on **IBM AD Connect for Mainframe**.

Using the z/OS User Authentication feature

1. After starting **IBM AD Build Client** or **IBM AD Build Configuration** with authentication **enabled**, the first operation you attempt and requires a mainframe connection will cause **IBM AD Build Client** or **IBM AD Build Configuration** to prompt you for your z/OS user credentials.
2. The credentials that you enter will persist for the duration of this instance of **IBM AD Build Client** or **IBM AD Build Configuration**. They will be used for all subsequent communication with **IBM AD Connect for Mainframe** until you stop the instance of **IBM AD Build Client** or **IBM AD Build Configuration**. After you restart **IBM AD Build Client** or **IBM AD Build Configuration**, you will be prompted again for your credentials. For more information, see Using the z/OS User Authentication feature in IBM AD Build Configuration and Using the z/OS User Authentication feature in IBM AD Build Client.

3. On the z/OS side, the only explicit indication of a successful login is the following message displayed in the JESMSGLG of the **IBM AD Connect for Mainframe** started task. It is also simultaneously displayed in the z/OS system log. This message is only displayed on systems running RACF as the z/OS security product. Some systems are configured to display this message only once per day at most.

   Example of a RACF message that is not part of the IBM AD product release.

   ```
   ICH70001I USERX  LAST ACCESS AT 09:16:31 ON SUNDAY, JUNE 6, 2021
   ```

   For security products other than RACF, they will generate their own unique messages following a successful user authentication. For more information, contact your site's security team.

Troubleshooting the z/OS User Authentication Configuration

- Make sure that both **IBM AD Build Client** and **IBM AD Connect for Mainframe Configuration** are configured to require user ID and password.
- Make sure that TLS is enabled on the z/OS Connection definition on **IBM AD Build Client Configuration**.
- Confirm with your z/OS security team that the user ID and password combination used as input are valid.
- Ensure that the user ID used to authenticate has read access to all the mainframe resources necessary for your IBM AD projects.
- For more detailed information, you can enable debug trace on **IBM AD Connect for Mainframe** by issuing the z/OS Modify command.

  ```
  F STC_NAME,DEBUGON
  ```

  Where STC_NAME is the started task name for the running instance of **IBM AD Connect for Mainframe**. The debug records are written to the STC IAYOUT file. You can search for error messages with a prefix of **IAYMF** that are generated at the time of your test. For more information, see "IBM AD Connect for Mainframe Messages" on page 528.

Enabling TCP/IP Connection to a Remote IWS Controller Address Space

This configuration procedure is required only if you need to connect to a remote IBM Workload Scheduler (IWS, previously known as TWS) controller address space that is running on another LPAR in your sysplex or in a remove z/OS instance.

IBM AD Connect for Mainframe can be configured to use the IWS program interface (PIF) that runs under the TCP/IP server for retrieving the information of applications and jobs from a remote IWS. The information refers to the IWS-defined applications, jobs, and their dependencies.

1. Ensure that the remote IWS controller is running. See the following checklist:

   The IWS TCP/IP server is configured and active.
   The IWS TCP/IP server is used for end-to-end processing. It communicates between the controller and the end-to-end domains. The IBM AD Connect for Mainframe server tasks use TCP/IP to handle communication to a remove IWS controller. Therefore, your z/OS systems must have the IWS TCP/IP server configured and active.

   See the following sample JCL statements for starting the IWS TCP/IP server address space:

   ```
   ******************************** Top of Data ****************************
   //TWC1S EXEC PGM=EQQSERVR,REGION=6M,TIME=1440
   //EQQMLIB DD DISP=SHR,DSN=TWS.V8R2M0.SEQQMSG0
   //EQQMLOG DD SYSOUT=*
   //EQQPARM DD DISP=SHR,DSN=EQQUSER.TWS01.PARM(SERP)
   //SYSMDUMP DD SYSOUT=*
   //EQQDUMP DD SYSOUT=*
   //*
   ```

See the following configuration parameters that are referenced by the IWS TCP/IP server address space. The **TCPIPJOBNAME**, **HOSTNAME**, and **SRVPORTNUMBER** parameters must be set for your site.

```
/*********************************************************************/
/* SERVOPTS: run-time options for SERVER KER
processor                                                 */
/*********************************************************************/
SERVOPTS   SUBSYS(OPCC)
/
*-------------------------------------------------------------------------------------------
-------------------*/
/* If you want to use Automatic Restart manager you must
specify:                               */
/*
ARM(YES)
                                        */
/
*-------------------------------------------------------------------------------------------
----------------- -*/
/* TCPOPTS : TCP/IP communications - local
definition                                        $CMDA */
/
*-------------------------------------------------------------------------------------------
----------------- -*/
            TCPOPTS  TCPIPJOBNAME('TCPIP')         == > TCPIP STACK
NAME
            HOSTNAME(9.30.114.131)  ==> Host IP Address
            SRVPORTNUMBER(350)      ==> Server port
Number
/*
CONNTIMEOUT(15)
                         */
/*
SSLLEVEL(FORCE)
                              */
/*          SSLKEYSTORE('wrkdir/ssl/
TWS.kdb')                                                                  */
/*          SSLKEYSTOREPSW('wrkdir/ssl/
TWS.sth')                                                        */
/*
SSLAUTHMODE(STRING)
                     */
/*
SSLAUTHSTRING('MYSTRING')
                  */
/* CALENDAR parameter is mandatory for server when using TCP/IP
*/
/*
server.
                                                 */
/
*-------------------------------------------------------------------------------------------
-----------------*/
INIT       CALENDAR(DEFAULT)
```

For detailed customization of the TCP/IP server, see *IBM Z Workload Scheduler Customization and Tuning*.

The IBM AD Connect for Mainframe user ID is defined to the security server of the remote LPAR where the IWS controller is running.

The user ID must remain the same. If the remote IWS controller is on another LPAR in the same sysplex and RACF sysplex data sharing is enabled, this user ID does not need to be defined again. The default group that the IBM AD Connect for Mainframe user ID is defined to must also be present on the remote LPAR. After the user ID is defined on the remote system, it must be given read access to the following resources:

The OPCC resource in the APPL class

If the APPL class is not set up, see chapter "Implementing security" in *IBM Z Workload Scheduler Customization and Tuning*. The following example shows how to grant user ID IAYV5108 read access. You can replace the user ID with the one that is assigned to IBM AD Connect for Mainframe at your site.

```
PERMIT OPCC ID(IAYV5108) ACCESS(READ) CLASS(APPL)
```

The IWS fixed resource AD

IBM AD Connect for Mainframe uses the AD resource of the IWS controller to retrieve job and application details. The IWS fixed resource that is named AD is coincidental and not related to IBM AD Connect for Mainframe.

If security is in place for restricted access to the resources, you must provide read access to the AD fixed resource. The AUTHDEF statement controls how IWS resource security is handled. The following example shows how to grant IBM AD Connect for Mainframe user ID IAYV5108 read access. IBMOPC is the IWS class that is defined in the AUTHDEF statement of the controller.

```
RDEFINE IBMOPC  AD UACC(NONE)
PERMIT AD ID(IAYV5108) ACCESS(READ) CLASS(IBMOPC)
SETROPTS RACLIST(IBMOPC) REFRESH
```

For setup of the default scheduler class IBMOPC and more details on the AUTHDEF statement or scheduler resources, see *IBM Z Workload Scheduler Customization and Tuning*.

2. APF authorize the IWS load-module library, or define the library in the linklist.

   IBM AD Connect for Mainframe requires the IWS library to be present on the LPAR as it users PIF modules to make the connection to the remote TCP/IP server. If you add the library to the IBM AD Connect for Mainframe started task JCL STEPLIB, you can APF authorize the IWS library. Update the active APF member (IEAAPF*nn* or PROG*nn*) to authorize the library.

   If you don't want to add the IWS library to the IBM AD Connect for Mainframe started task JCL STEPLIB and APF authorize the IWS library, you can define the library in the linklist.

3. Configure local OPCC.

   For local OPCC configuration, the IBM AD Connect for Mainframe started task JCL needs the following DD statements:

   //EQQMLIB DD DISP=SHR,DSN=TWS951.SEQQMSG0
   This line can be changed if the IWS message library has a different name on your site. If a user library is defined for IWS messages, this user library can be placed first in the EQQMLIB DDname concatenation, and the two libraries names might be concatenated.

   //EQQDUMP DD DISP=SHR,DSN=IAY.USER.TWS.EQQDUMP
   The EQQDUMP DDname refers to an IWS dump data set that is created to capture IWS dump. This data set must be created with record format VBA, record length 84, and block size 3120. 10 primary tracks are sufficient, but you might allocate more.

   //EQQMLGnn DD DISP=SHR,DSN=IAY.USER.TWS.MLOG.TASK*nn*

   Creates MAXTASK data sets named AY.USER.TWS.MLOG.TASK*nn* as sequential files with record format VBA, record length 125, and block size 1632. *nn* goes from 01 to the number of tasks that are specified in MAXTASK. 10 primary tracks are sufficient.

4. Configure remote OPCC.

   For remote OPCC configuration, the IBM AD Connect for Mainframe started task JCL needs two additional DD statements: EQQYPARM and EQQMLOG.

   //EQQYPARM DD DISP=SHR,DSN=IAY.AD.PARMLIB(PIFPARM) : -
   The PDS or sequential file for the EQQYPARM DDname can be a member of a partitioned data set or a sequential file. This line represents the EQQYPARM statement required for the remote connection with the SERVER task that is running on the remote machine.

   The EQQYPARM statement contains an initialization statement: INIT. The INIT statement overrides the values that are set by the INTFORTS statement in the EQQYPARM statement. The most important settings are SUBSYS, REMHOSTNAME, REMPORTNUMBER, and DUBPROC. See the following EQQYPARM sample:

```
/*********************************************************************/
/* INIT STATEMENT FOR PIF PROGRAMS (EQQYPARM DD)   */
/*********************************************************************/
 INIT  CALENDAR(DEFAULT)
OIWSNAME(IGNORE)
SUBSYS(OPCC)                                        /*Remote TWS  controller
```

```
Subsystem name*/
TRACE(0) CWBASE(00) HIGHDATE(711231)
REMHOSTNAME(9.30.114.131)                          /* Remote MAINFRAME IP STACK
*/                    REMPORTNUMBER(350)         /* PORTNUMBER  IN SERVOPTS  on remote
LPAR */                    DUBPROC(Y)                    /*should not be changed or
deleted for AD connect for MF*/
```

// EQQMLOG DD SYSOUT=*

The EQQMLOG DDname is the message log data set, and it must be defined in the IBM AD Connect for Mainframe started task.

The message log data set can be written to SYSOUT or a data set. The data control block (DCB) for this data set is defined by IWS as follows:

```
EQQMLOG Dataset DCB attributes (RECFM=VBA,LRECL=125,BLKSIZE=1632)
```

The following example shows the IBM AD Connect for Mainframe procedure JCL setup for an IWS remote connection.

```
//IAYV5107 PROC RVER=IAYV5108,
//*    BVER=IAYV510,
//*    TVER=IAYV5108,
//     PGMNAME=IAYLISPR,
//     CONNNAME=IAYCONN,
//     TCPAS=TCPIP,                     --> TCP/IP ADDRESS SPACE NAME
//     PORT=05108,                      --> TCP/IP PORT NUMBER
//     HOSTCP=00037,                    --> HOST CODE PAGE (CCSID)
//     CLNTCP=00850,                    --> CLIENT CODE PAGE (CCSID)
//     MAXTK=5,                         --> MAX TASK FOR THIS RUN
//     RXI=N,
//     DEB=L                         --> DEBUG MODE THIS RUN (Y/N/L/S/T)
//LISTEN EXEC PGM=&PGMNAME,REGION=0M,
//  PARM='&TCPAS,&PORT,&MAXTK,&CONNNAME,&DEB,&HOSTCP,&CLNTCP,N,&RXI'
//STEPLIB  DD  DISP=SHR,DSN=&RVER..LOADLIB
//* -- TWS APF LOAD LIB BELOW
//      DD  DISP=SHR,DSN=TWS.V9R5M0.SEQQLMD0 ==> Workload Scheduler
LOADLIB
//IAYINTR  DD  SYSOUT=(*,INTRDR)
//IAYINT01 DD  SYSOUT=(*,INTRDR)
//IAYINT02 DD  SYSOUT=(*,INTRDR)
//IAYINT03 DD  SYSOUT=(*,INTRDR)
//IAYINT04 DD  SYSOUT=(*,INTRDR)
//IAYINT05 DD  SYSOUT=(*,INTRDR)
//* <-- IAY INTERNAL OUTPUT DDS
//CPEOUT   DD  SYSOUT=*
//IAYOUT   DD  SYSOUT=*
//IAYERR   DD  SYSOUT=*
//CPEOUTRS DD  SYSOUT=*
//CPEOUTRQ DD  SYSOUT=*
//CPESNAP  DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//* <-- IAY INTERNAL REXX INTEGRATION DD
//SYSEXEC  DD  DISP=SHR,DSN=&RVER..SAMPLIB
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY,BLKSIZE=80
//SAMPLJCL DD  DISP=SHR,DSN=&RVER..SAMPLIB
//IAYIRXPC DD  DISP=SHR,DSN=&RVER..SAMPLIB(IAYINI00)
//***TWS
//EQQYPARM DD DISP=SHR,DSN=IAYV510.AD.PARMLIB(PIFPARM) ==> Parameter
//*
 file for remote connection
//EQQMLIB DD DISP=SHR,DSN=TWS.V9R5M0.SEQQMSG0 ==> Workload
//*
            Scheduler MSGLIB
//EQQDUMP DD SYSOUT=*
//EQQMLOG DD SYSOUT=*                        ==> EQQMLOG is
mandatory
//EQQMLG01 DD DISP=SHR,DSN=IAYV5108.TWS.MLOG.TASK01
//EQQMLG02 DD DISP=SHR,DSN=IAYV5108.TWS.MLOG.TASK02
//EQQMLG03 DD DISP=SHR,DSN=IAYV5108.TWS.MLOG.TASK03
//EQQMLG04 DD DISP=SHR,DSN=IAYV5108.TWS.MLOG.TASK04
//EQQMLG05 DD DISP=SHR,DSN=IAYV5108.TWS.MLOG.TASK05
```

Chapter 6. Managing Security in IBM AD

Securing Apache ZooKeeper SSL connections

Setting up IBM AD ZooKeeper to run in Mixed Mode

Before you begin:

- Make sure that a Java™ Runtime Environment is installed on your machine and that the **JAVA_HOME** variable is defined in the Environment variables. For more information, see the “Java Runtime Environment ” on page 9 section.
- You can use any keytool utility that is provided with JDK on both Linux™ and Windows™ to generate self-signed certificates. In this scenario, **OpenSSL** tool from **Cygwin** is used to generate a self-signed key certificate for IBM AD Build Client.

  Note: In case that you don't want to use a self-signed certificate, make sure that a certificate authority (CA) issues a signed certificate.

By manually enabling the IBM AD ZooKeeper instance to run in mixed mode, the server will respond on both types of requests:

- Requests that come on non-authenticated channel.
- Requests that come on encrypted channel.

The main advantage in having IBM AD ZooKeeper set up in mixed mode is that you can easily access the IBM AD ZooKeeper server remotely by using the Transport Layer Security (TLS) protocol.

The TLS protocol is a client or server cryptographic protocol. It is based on the earlier Secure Sockets Layer (SSL) specifications that are developed by Netscape Corporation for securing communications that use Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. The TLS and SSL protocols are designed to run at the application level. Therefore, an application must be designed and coded to use TLS/SSL protection.

For more information, see the Communication using the Netty framework section of *ZooKeeper Administrator's Guide*.

The IBM AD ZooKeeper server needs to be configured to work on separate ports. The default ports are:

- **2181** - non-authenticated mode.
- **2281** - authenticated mode (TLS enabled).

By default, the IBM AD ZooKeeper server is installed by IBM Application Discovery application and runs in non-authenticated mode. To configure IBM AD ZooKeeper as a server with TLS support, you need to perform the following two major steps:

1. Generate a self-signed key certificate for IBM AD ZooKeeper.
2. Activate the IBM AD ZooKeeper Server to use the generated certificates .

Generate a self-signed key certificate for IBM AD ZooKeeper

Important: Each server must have a keystore containing a key certificate (private key + public certificate). As mentioned before, the key certificate can be self-signed or signed by a certificate authority (CA).

1. At your convenience, use any tool to generate the certificates in a dedicated location. For example, `C:\certificates`.

© Copyright IBM Corp. 2010, 2022

2. Use the Java™ `keytool` utility with the following input to generate private and public key (key pair).

```
keytool -genkeypair -alias "<machine name/IP>" -keyalg RSA -keysize 2048 -dname "cn=<machine
name/IP>" -keypass password -keystore keystore.jks -storepass password
```

   Note: The alias (-alias) and the distinguished name (-dname) must match the hostname of the machine that it is associated or the hostname verification fails.

   The generated key is in the specified keystore.

3. Export the certificate.

```
keytool -exportcert -alias "<machine name/IP>" -keystore "keystore.jks" -file "adzk.cer"
-storepass "password"
```

4. Create the SSL truststore JKS containing the certificates of all ZooKeeper instances.

```
keytool -importcert -alias "<machine name/IP>" -file adzk.cer -keystore truststore.jks
-storepass password
```

   Note: Choose **Yes** when you are asked about trusting this certificate.

5. Check the folder where the certificates were created. Both `keystore` and `truststore` files should be there.

Activate the IBM AD ZooKeeper Server to use the generated certificates

1. Access **IBM Application Discovery Configuration Service** from **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, go to the **Dashboard** tab, click the menu button of **Configuration Server**, and select **Stop Service**.
2. Go to the `<IBM ADDI Installation Folder>\IBM Application Discovery Configuration Service\conf` folder, open the `zoo.cfg` file by using a text editor, uncomment and enter the desired values for the following parameters. Example:

```
secureClientPort=2281
serverCnxnFactory=org.apache.zookeeper.server.NettyServerCnxnFactory
ssl.keyStore.location=c:\\certificates\\keystore.jks
ssl.keyStore.password=password
ssl.trustStore.location=c:\\certificates\\truststore.jks
ssl.trustStore.password=password
```

   Note: You can enter an encrypted **Keystore Password**. For more information, see “Obtaining an encrypted keystore password” on page 209.

   Where:

   - `secureClientPort` - expects the port on which the secure channel communicates.
   - `serverCnxnFactory` - for a successful authentication with TLS do not modify the value of this parameter.
   - `ssl.keyStore.location` - expects the location on disk where the keystore was stored.
   - `ssl.keyStore.password` - expects the keystore's password.
   - `ssl.trustStore.location` - expects the location on disk where the truststore was stored.
   - `ssl.trustStore.password` - expects the truststore's password.

3. Once IBM AD Configuration Server is configured to run in mixed mode, make sure to block the 2181 port on the Firewall for the incoming traffic. For more information, see “Block the incoming traffic for the non-authenticated port of IBM AD Zookeeper” on page 205.

Block the incoming traffic for the non-authenticated port of IBM AD Zookeeper

In the following section, you can find all the necessary information on how to configure the Firewall for both Windows and Linux to restrict incoming traffic on the IBM AD ZooKeeper's port that is not running in authenticate mode.

IBM AD ZooKeeper server used by IBM Application Discovery and Delivery Intelligence (IBM ADDI) and IBM Application Discovery (IBM AD) can run in non-authenticated way or mixed mode. For more information, see "Setting up IBM AD ZooKeeper to run in Mixed Mode" on page 203.

When IBM AD ZooKeeper is configured in mixed mode, it still accepts non-authenticated requests. You can filter out those calls if the requests come from outside the machine where IBM AD ZooKeeper is running.

Check the following procedures for each operating system on how to close IBM AD ZooKeeper non-authenticated port.

How to close the IBM AD ZooKeeper non-authenticated port on Windows 10

Note: The following instructions apply to the Windows 10 distribution and to the default Windows Defender Firewall. In case that you are using any other firewall, the same major steps apply.

To close the IBM AD ZooKeeper non-authenticated port, perform the following steps:

1. On the Windows taskbar, click **Start** > **Control Panel** > **System and Security** > **Windows Defender Firewall**.
2. Click **Advanced Settings**.
3. Make sure that **Windows Defender Firewall** is **on**.
4. In the *Windows Firewall with Advanced Security* window, click **Inbound Rules** > **New Rule**.

   Note: The inbound rules are handling the requests that are coming from other machines.
5. On the *Rule Type* page, click **Port** and then click **Next**.
6. On the *Protocol and Ports* page, click **Specific local ports** and then type the 2181 port number that is used by IBM AD ZooKeeper to communicate on non-authenticated mode.
7. Click **Next**.
8. On the *Action* page, click **Block the connection** and then click **Next**.
9. On the *Profile* page, select all profiles and click **Next**.
10. On the *Name* page, type a name for the rule and click **Finish**.

    Example: `Block ZK 2181 port`
11. In the *Windows Firewall with Advanced Security* window, click **Inbound Rules** > **New Rule**.

    Note: The inbound rules are handling the requests that are coming from other machines.
12. On the *Rule Type* page, click **Port** and then click **Next**.
13. On the *Protocol and Ports* page, click **Specific local ports** and then type the 2281 port number that is used by IBM AD ZooKeeper to communicate on TLS.
14. Click **Next**.
15. On the *Action* page, click **Allow the connection** and then click **Next**.
16. On the *Profile* page, select all profiles and click **Next**.
17. On the *Name* page, type a name for the rule and click **Finish**.

    Example: `Allow ZK 2281 port`
18. Once the IBM AD ZooKeeper non-authenticated port is closed, make sure to check the "TCP Port Requirements and Firewall Exceptions" on page 20 page. In order for IBM AD to run correctly you need to enable all other components ports.

How to enable the IBM AD ZooKeeper to run in authenticated mode (TLS enabled) on Linux Ubuntu 18.04

Note:

- The following instructions apply to the Linux Ubuntu 18.04 distribution. The same procedure applies when using another Linux OS where IBM ADDI or IBM AD is installed.
- The Uncomplicated Firewall (UFW) interface is used to simplify the process of configuring a firewall. By default, UFW is installed on Linux Ubuntu. If for some reason, it has been uninstalled, you can install it by using the `sudo apt install ufw` command.

By default, Linux firewalls are blocking all the incoming traffic. You need to enable the firewall and allow incoming traffic to 2281 port number (default port of the TLS communication for IBM AD ZooKeeper). The other ports are going to be blocked for any incoming traffic. In case that another port needs to be opened, a new rule needs to be defined on the firewall.

To enable the firewall and allow incoming traffic to 2281 port number, perform the following steps:

1. Check the UFW status by using the `sudo ufw status verbose` command.
2. In case that the firewall is disabled, use the `sudo ufw enable` command.
3. Make sure that the firewall is enabled by using the `sudo ufw status verbose` command.
4. Assuming that the 2281 port number is used by IBM AD ZooKeeper to communicate on TLS, add it to the firewall by using the `sudo ufw allow 2281` command.
5. Check the UFW status by using the `sudo ufw status verbose` command.

   Note: As a result, two new rules should be displayed. One is for IPV6 the other one for IPV4.
6. Once IBM AD ZooKeeper IBM AD ZooKeeper runs in authenticated mode make sure to check the “TCP Port Requirements and Firewall Exceptions” on page 20 page. In order for IBM AD to run correctly you need to enable all other components ports.

Enabling encryption channel between IBM AD Build Client and IBM AD ZooKeeper

Before you begin:

- Make sure that a Java™ Runtime Environment is installed on your machine and that the **JAVA_HOME** variable is defined in the Environment variables. For more information, see the “Java Runtime Environment ” on page 9 section.
- You can use any keytool utility that is provided with JDK on both Linux™ and Windows™ to generate self-signed certificates. In this scenario, **OpenSSL** tool from **Cygwin** is used to generate a self-signed key certificate for IBM AD Build Client.

  Note: In case that you don't want to use a self-signed certificate, make sure that a certificate authority (CA) issues a signed certificate.

For earlier versions, the communication between IBM AD Build Client and IBM AD ZooKeeper is unencrypted socket session. Beginning with version IBM AD V6.0.0 interim fix 1, you can configure IBM AD Build Client to enable Transport Layer Security (TLS) connection.

The TLS protocol is a client or server cryptographic protocol. It is based on the earlier Secure Sockets Layer (SSL) specifications that are developed by Netscape Corporation for securing communications that use Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. The TLS and SSL protocols are designed to run at the application level. Therefore, typically, an application must be designed and coded to use TLS/SSL protection.

By default, the IBM AD Build Client runs in non-authenticated mode. To configure IBM AD Build Client with TLS support, you need to perform the following two major steps:

1. Generate a self-signed key certificate for IBM AD Build Client.

2. Activate IBM AD Build Client to use the generated certificates.

Generate a self-signed key certificate for IBM AD Build Client

Important: Each server must have a keystore containing a key certificate (private key + public certificate). As mentioned before, the key certificate can be self-signed or signed by a certificate authority (CA).

1. Use the **OpenSSL** tool from **Cygwin** to generate the certificates in a dedicated location. For example, `C:\certificates`.
2. To generate the `key.pem` and `cliCert.pem`, run the following command.

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cliCert.pem
```

   Note: A password will be requested when you run the above command to encrypt the client's private key. This password will be required when you will edit the `zoo.ini` configuration file.
3. To import the certificate, run the following command.

```
keytool -importcert -alias zoo_c_client -file cliCert.pem -keystore truststore.jks
-storepass password -deststoretype jks
```

4. Check the folder where the certificates were created. Both `key.pem` and `cliCert.pem` files should be there.

Activate IBM AD Build Client to use the generated certificates

1. Make sure that IBM AD ZooKeeper is configured as a server with TLS support. For more information, see the "Securing Apache ZooKeeper SSL connections" on page 203 section.
2. On the machine where **IBM AD Build Client** is installed, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Build Client/Bin/Release` and make sure that the `zoo.ini` is present. If the `zoo.ini` file is not present in the `/Release` folder, go to `<IBM ADDI Installation Folder>/IBM Application Discovery Build Client/Bin/Release/Samples` and copy the `zoo.ini` file in the `/Release` folder. Open the `zoo.ini` file by using a text editor and enter the desired values for the parameters that are detailed below. Example:

```
[SSL/TLS]
;SSL/TLS initialization parameters for Zookeeper
;Allowed values to enable secure communication: y or Y (case insensitive). Anything else
means disable secure communication.
zoo_enable_secure_communication=y
zoo_secure_port=2281

;Settings for IBM Build Client & Configuration:
;Fully qualified file name for the 'zookeeper' server certificate, E.g.:
'C:\Securitycerts\certZoo\9.20.128.30\zoo.cer'
zoo_server_certificate=C:\Certificates\certZoo\zoo.cer
;Fully qualified file name for the 'zookeeper' client certificate, in .pem format, E.g.:
'C:\Securitycerts\tmp\cliCert.pem'
zoo_client_certificate=C:\Certificates\clientCertZoo\cliCert.pem
;Fully qualified file name for the 'zookeeper' client private key, in .pem format, E.g.:
'C:\Securitycerts\tmp\key.pem'
zoo_client_privateKey=C:\Certificates\clientCertZoo\key.pem
;Pass phrase asked when generating the private key:
zoo_client_privateKeyPwd=password

;Settings for java compilers:
zoo_java_client_keystore=C:\Certificates\certZoo\keystore.jks
zoo_java_client_keystorePwd=password
zoo_java_client_truststore=C:\Certificates\certZoo\truststore.jks
zoo_java_client_truststorePwd=password

;end of [SSL/TLS] section
```

Detailed information about IBM Build Client & Configuration settings

- `zoo_enable_secure_communication` - set to **Y** to enable TLS connection.
- `zoo_secure_port` - expects the port number that matches to the one that has been assigned for **IBM AD ZooKeeper**. Example: **2281**.

- `zoo_server_certificate` - expects the fully qualified file name of the certificate.

  Important: Make sure that the **IBM AD Build Client's** certificate matches to the one that has been generated for **IBM AD ZooKeeper**.
- `zoo_client_certificate` - expects the fully qualified file name of `cliCert.pem`.
- `zoo_client_privateKey` - expects the fully qualified file name of `key.pem`.
- `zoo_client_privateKeyPwd` - expects the password that was selected when the certificate was generated.

Detailed information about Java compilers settings

Important:

Make sure that the `keystore.jks` and `trustore.jks` files are physically present on the machine where **IBM AD Build Client** is installed and configured. You may use the `keystore.jks` and `trustore.jks` files that were generated for **IBM AD ZooKeeper** if both **IBM AD Build Client** and **IBM AD ZooKeeper** are installed on the same machine. For more information, see "Generate a self-signed key certificate for IBM AD ZooKeeper" on page 203.

- `zoo_java_client_keystore` - expects the fully qualified file name of the keystore.
- `zoo_java_client_keystorePwd` - expects the keystore's password.
- `zoo_java_client_truststore` - expects the fully qualified file name of the truststore.
- `zoo_java_client_truststorePwd` - expects the truststore's password.

3. Start **IBM AD Build Client**.

   Note: If you select the **Open Project** option the list of projects from the server machine should be available.

Enabling encryption channel between IBM AD Analyze Client and IBM AD ZooKeeper

Before you begin:

- Make sure that a Java™ Runtime Environment is installed on your machine and that the **JAVA_HOME** variable is defined in the Environment variables. For more information, see the "Java Runtime Environment " on page 9 section.
- You can use any keytool utility that is provided with JDK on both Linux™ and Windows™ to generate self-signed certificates. In this scenario, **OpenSSL** tool from **Cygwin** is used to generate a self-signed key certificate for IBM AD Analyze Client.

  Note: In case that you don't want to use a self-signed certificate, make sure that a certificate authority (CA) issues a signed certificate.

For earlier versions, the communication between IBM AD Analyze Client and IBM AD ZooKeeper is unencrypted socket session. Beginning with version IBM AD V6.0.0 interim fix 1, you can configure IBM AD Analyze Client to enable Transport Layer Security (TLS) connection.

The TLS protocol is a client or server cryptographic protocol. It is based on the earlier Secure Sockets Layer (SSL) specifications that are developed by Netscape Corporation for securing communications that use Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. The TLS and SSL protocols are designed to run at the application level. Therefore, typically, an application must be designed and coded to use TLS/SSL protection.

By default, the IBM AD Analyze Client runs in non-authenticated mode. To configure IBM AD Analyze Client with TLS support, you need to perform the following steps:

1. Make sure that IBM AD ZooKeeper is set up in mixed mode. For more information, see the "Securing Apache ZooKeeper SSL connections" on page 203 section.

2. Edit the `eclipse.ini` configuration file and add the following lines, in the `-vmargs` section. Avoid blank lines in the `-vmargs` section. Example:

```
-Dzookeeper.client.secure=true
-Dzookeeper.ssl.keyStore.location=C:\Certificates\keystore.jks
-Dzookeeper.ssl.keyStore.password=password
-Dzookeeper.ssl.trustStore.location=C:\Certificates\truststore.jks
-Dzookeeper.ssl.trustStore.password=password
```

Important:

- Make sure that the `keystore.jks` and `trustore.jks` files are physically present on the machine where **IBM AD Analyze Client** is installed and configured. You may use the `keystore.jks` and `trustore.jks` files that were generated for **IBM AD ZooKeeper** if both **IBM AD Analyze Client** and **IBM AD ZooKeeper** are installed on the same machine. For more information, see "Generate a self-signed key certificate for IBM AD ZooKeeper" on page 203.
- You can enter an encrypted **Keystore Password**. For more information, see "Obtaining an encrypted keystore password" on page 209.

Where:

- `-Dzookeeper.client.secure` - set to **`true`** to enable TLS connection.
- `-Dzookeeper.ssl.keyStore.location` - expects the location on disk where the keystore was stored.
- `-Dzookeeper.ssl.keyStore.password` - expects the keystore's password.
- `-Dzookeeper.ssl.trustStore.location` - expects the location on disk where the truststore was stored.
- `-Dzookeeper.ssl.trustStore.password` - expects the truststore's password.

3. Start **IBM AD Analyze Client**.
4. Go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Environment settings** and enter the following information:
   - **Host** - type the hostname or the IP address of the machine where **IBM AD ZooKeeper** is installed.
   - **Port**- type the 2281 port number that is used by **IBM AD ZooKeeper** to communicate on TLS.
   - **Unique id** - type the unique id assigned by **IBM AD Configuration Server** to the current environment.
   - **Name** - type the name of current environment, as defined in **IBM AD Configuration Server**.
5. Click **Apply and Close** and restart **IBM AD Analyze Client**.
6. When **IBM AD Analyze Client** starts, the list of the mainframe projects is empty. To have the list of mainframe projects available, it is necessary to use the **Get project list** contextual-menu option by right-clicking in the **Explore projects view**.

Obtaining an encrypted keystore password

This page explains how to encrypt a keystore password.

When HTTPS is enabled in IBM AD Configuration Server

Note: Make sure that **Hypertext transfer protocol secure (HTTPS)** is set as a default connection protocol in IBM AD Configuration Server. For more information, see "Configuring Security Settings" on page 42.

1. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Configuration Service\conf`.
2. Open the `jvmargs.txt` file by using a text editor.
3. Copy the encrypted password value from `-Djavax.net.ssl.keyStorePassword=<ENCRYPTED_PASSWORD>`.

When HTTPS is disabled in IBM AD Configuration Server

1. Open a new Command Prompt on Windows or a new Terminal on Linux.
2. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Configuration Service\conf`.
3. Execute the following command:
   - For Windows

     ```
     - securityUtility.bat <password>
     ```

   - For Linux

     ```
     - ./securityUtility.sh <password>
     ```

     Note: The script file might need the required permissions to be executed. To grant access execute:

     ```
     chmod +x securityUtility.sh
     ```

4. The encrypted password is displayed on the screen.

Other security recommendations

IBM AD Build Client - Enable SMB2 Encryption to avoid any possible sniffing of the network packages

Scenario

IBM AD Build Client requests and receives files via SMB2 in plain text. When transferring files between machines the network connection might get sniffed.

Fix recommendation

When installing the IBM AD application, you can enable **SMB Encryption** with **Windows PowerShell**. For more information, see **SMB security enhancements**.

Note: Make sure that **SMB Encryption** is enabled on the machine where **IBM AD Build Client** is installed.

IBM AD Build Client - DLL Hijacking

How to prevent DLL Hijacking

DLL hijacking isn't a vulnerability. It is just an exploitation technique for getting code execution in the context of an application.

An exploitation technique on its own is useless if there isn't a vulnerability such as weak folder permissions or a privileged file operation to abuse. In order to minimize this risk, system administrators need to pay extra attention and change the permissions on the **IBM AD Build Client's** installation folder to allow only the appropriate users to have administrative rights on it.

Securing Access on IBM AD Services Endpoints

By default, the endpoints available on the ports of **IBM AD Audit Service**, **IBM AD Cross Applications Service**, **IBM AD File Service**, **IBM AD Search Service**, **IBM AD Mainframe Projects Service**, **IBM AD Manual Resolutions Service**, **IBM AD WebSphere® Liberty Profile Service**, and **Authentication Server (DEX)** are publicly available inside the domain, where no access control is enforced.

You can enable **Hypertext transfer protocol secure (HTTPS)** as a default connection protocol as follows:

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security**. The **Servers and security** settings page is displayed.
2. Select **HTTP Secure (https)** as a default connection protocol.

   Important: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`), a private key for the certificate (`.key`), and the keystore file that needs to have one of the following extensions: `.jks`, `.keystore`, `.pfx`, `.p12`, or `.ks`.
3. Drag and drop the three required files or click to browse.
4. After all files are successfully uploaded enter the **Keystore Password**.
5. Click **Save**
6. If IBM AD Audit has been configured, additional configurations are required for IBM AD Build Client and IBM AD Build Configuration. For more information, see Configuring IBM AD Build Client and IBM AD Build Configuration to send Audits events securely.

Using the Authentication and Authorization Feature

To be authenticated and authorized to have the list of the projects displayed in IBM AD Analyze Client, you need to perform the following actions:

1. “Configure Authentication Server (DEX)” on page 211
2. “Authenticate in IBM AD Analyze Client” on page 211

Configure Authentication Server (DEX)

The current implementation of the **Authentication and Authorization** feature, is based on **OAuth2** authorization framework, and needs to have available an **Authentication Server (DEX)**.

The provided **Authentication Server** is based on DEX and provides an authentication solution, which connects through **LDAP** to **Secure Storage**. For more information about DEX, go to the DEX documentation.

Follow the configuration steps that are needed to have up and running **Authentication Server (DEX)**:

1. Configure the parameters that are present in the dex.yaml file
2. Make Authentication Server (DEX) available in IBM AD Configuration Server
3. Restart Authentication Server (DEX)

Authenticate in IBM AD Analyze Client

If the **Authentication and Authorization** feature is enabled, you need to be authenticated in **IBM AD Analyze Client** to have the list of the projects displayed.

To be authenticated and authorized, follow the instructions that are present on the Authorization and Authentication page.

Restrictions that can be applied

- **On projects** - by selecting whether a workspace is public, closed, or available only for specified user groups, you can restrict the number of projects that a user can see. If you select the **Access Control List (ACL)** option, you can then select the user groups that have access to the current workspace. For more information about user access to workspaces, see Managing workspaces' access rights.
- **On Source Files** - by defining which **User Groups** can access the related mapping folders in **IBM AD File Service**, you can restrict the source files that a user can access through **IBM AD Analyze Client**. For more information, see Configuring IBM AD File Service.

Mask usernames in IBM AD Configuration Server

By default, all usernames are visible in IBM AD Configuration Server. For security reasons, you can hide part of the username in IBM AD Configuration Server by modifying the `audit.properties` configuration file.

1. Go to `<IBM ADDI Installation Folder>/IBM AD Web Services/wlp/usr/servers/ad_server/conf.audit/`, open the `audit.properties` file, and add the following line.

   ```
   com.ibm.ad.audit.username.mask=true
   ```

2. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, go to the **Dashboard** tab, click the menu button of **Audit Service**, and select **Restart Service**.

   Note: Wait until the services are restarted, this can take a few minutes to complete.

3. After **Audit Service** has been restarted, go to the **Audit** tab, where all usernames should be masked.

Chapter 7. Managing Application Projects with IBM AD Build

The objective is to provide the information that is needed to use IBM AD Build Client, and to understand the capabilities.

A description of the following sequence of steps and procedures that are typically followed to set up and analyze a system, are described as follows.

1. Setup
   - Create a project.
   - Add files to the project.
   - Update project resources.
2. Analysis
   - Collect information on an application (called a 'Build') and store the results in the repository.
   - Make and integrate the current version of a project resource into the built project.
   - Search in project for a specific project resource.

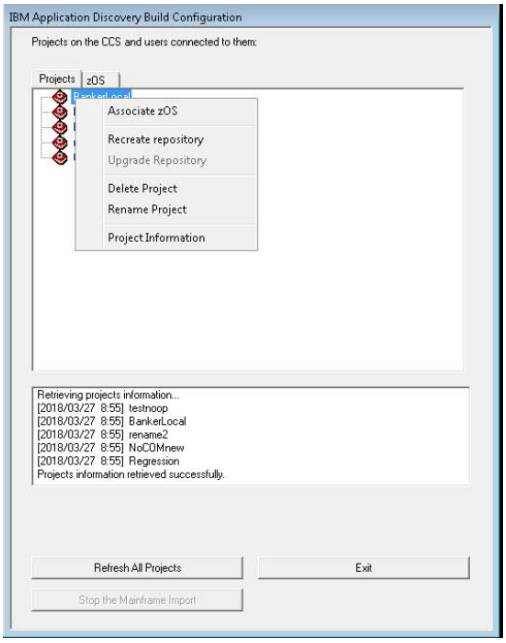
Using IBM AD Build Configuration

You can perform the following actions in **IBM AD Build** using the Configuration Tool:

- View existing projects.
- Delete a project.
- Rename a project.
- Re-create a repository.
- Upgrade a repository.
- Display the users who are currently using a project.
- Create and configure a z/OS connection to a remote computer.

To open **IBM AD Build Configuration**, click **Start** > **Programs** > **IBM Application Discovery Build Client** > **IBM Application Discovery Build Configuration**.

© Copyright IBM Corp. 2010, 2022



Viewing Project Information

To view project information, follow these steps:

1. In the **IBM Application Discovery Build Configuration** window - right-click on the project and choose **Project Information**.
2. A window is displayed showing the **Project Path**, **Project Database Connection String**, **Project Creation Information**, and **Authentication Information**.
3. If you are working in multi-user mode, under each project a list of users who are currently logged in is displayed.

Deleting a Project

Deleting a project can be done in two ways:

In the **IBM Application Discovery Build Configuration** window.

a) Select the project to be deleted from the list of projects.

b) Right-click to display the menu and select **Delete Project**.

c) You are asked to confirm the deletion request.

d) If other users are logged in to the project, a warning message appears listing all the users who are connected to the project.

Renaming a Project

To rename a project, goto the **IBM Application Discovery Build Configuration** window and follow these steps:

1. Select the project to be renamed.
2. Right-click to display the menu, select **Rename project**.
3. A window is displayed waiting you to confirm the operation.
4. The project with the new name is displayed in the projects list.

   Note: It is highly recommended to perform a build after you rename the IBM AD project. For more information, see “Building Projects” on page 264.

Recreate a Repository

To re-create a repository in case the current repository was deleted or got corrupted, follow these steps:

Note: Recreate Repository is not available for projects that are attached to an existing Db2 on z/OS database when they are created.

1. From **IBM AD Build Configuration** window, right-click the selected project and choose **Recreate Repository**.
2. A warning message appears, waiting for you to confirm the recreation. After you confirm the recreation, the repository is re-created, and a full build can be done by using an **IBM Application Discovery Build Client** to allow analysis for the project.
3. If other users are logged in to the project, a warning message informs the other users about the operation about to be run.

Upgrade a Repository

If other users are logged in to the project, a warning message informs the other users about the operation about to be run.

Note: Upgrade repository feature was enhanced. If you need to perform changes on AD project in the AD Build, you must upgrade the repository first.

1. To upgrade the repository for a single project, follow these steps:
   a) In the **IBM Application Discovery Build Configuration** window, right-click the selected project and choose **Upgrade repository** if available.
   b) A warning message appears, waiting for you to confirm the upgrade. After you confirm, the repository is upgraded to the current version.
   c) A warning message appears, waiting for you to confirm the upgrade. After you confirm, the repository is upgraded to the current version.
2. To upgrade the repository for a list of projects, follow these steps:
   a) First, create a text file by specifying the list of projects to be included in the repository upgrade operation.

      Important: Each project name must appear on a separate line in the text file.
   b) Open the command prompt and enter the following command:

```
C:\Program Files\IBM Application Discovery Build Client\Bin\Release\IBMApplicationDiscoveryBuildClient.exe"/ ru <fully qualified LOG file name>
<fully qualified projects file name>
```

      Where,

      - <fully qualified LOG file name> is, the log file that is created detailing the results of the upgrade operation.
      - <fully qualified projects file name> contains the file name with the projects to be upgraded, one per line.

      If <fully qualified projects file name> is not present, all projects are upgraded.

      Tip: This operation runs in the background.

Stop the Mainframe Import

This function is used to cancel the library scanning, query environment and get files operations.

If other users are logged in to the project, a warning message informs the other users about the operation about to be run.

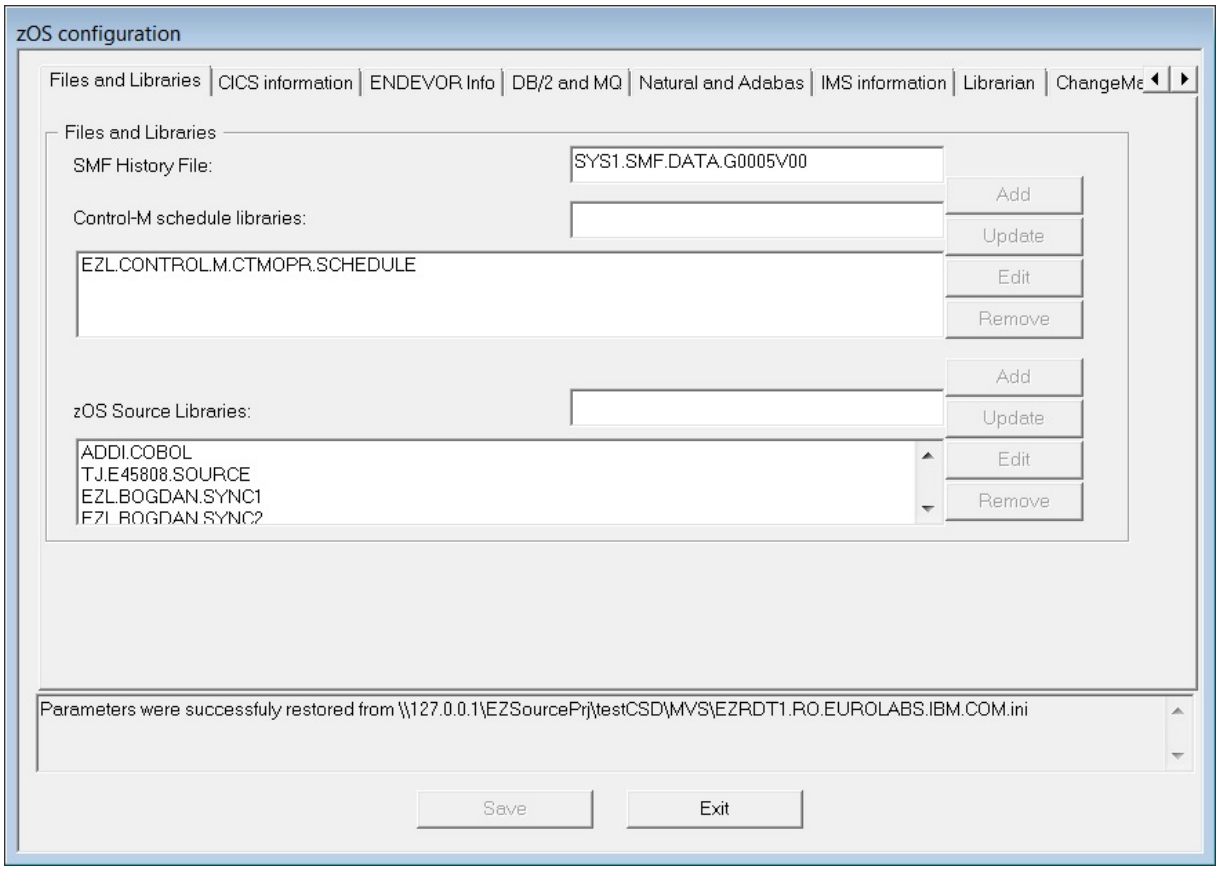
Configuring the z/OS Connection

You can use this option to specify different settings for the **z/OS Connection** and the names of the libraries that contain the resources you want to import in your project. You can enter all the data that you consider relevant in a dedicated tab, for each type of resource.

Important: Authentication with proper credentials through Carma/Endevor is enforced before one can read/obtain any source file, regardless of method used for obtaining these sources:

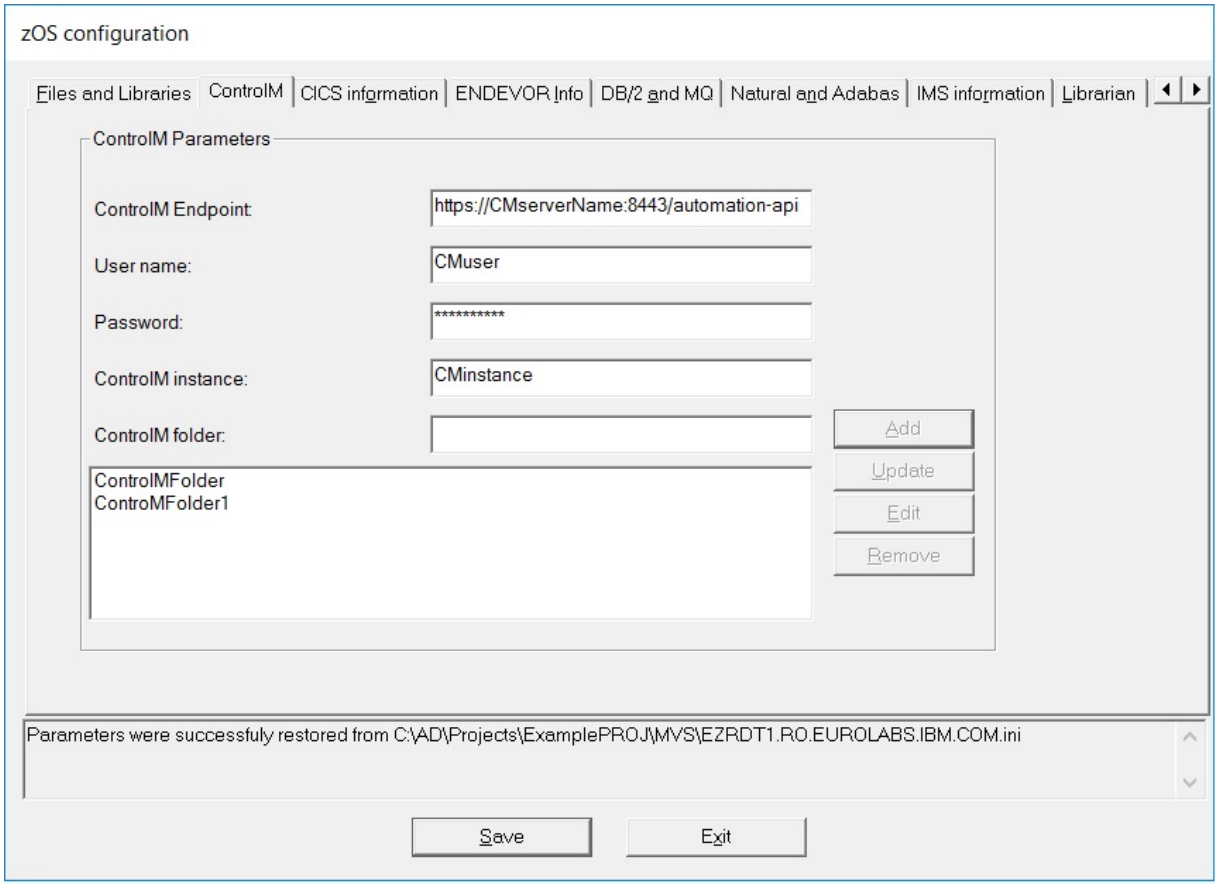
1. By downloading from mainframe (Endevor)
2. Otherwise provided. These sources can only be viewed if they are shared over the network. Read only access is required.

1. Goto **IBM Application Discovery Build Configuration**, **Projects** tab and select the project then the corresponding z/OS node, then right-click to display the menu. Select **Configure connection** to display the following window.



2. Complete the settings or the names of the libraries that contain the resources you want, according to the type of resource you need to import.
3. To add a library, enter its name in the corresponding field, then click **Add**. The name of the new library is displayed in the library list. Use the **Update**, **Edit**, and **Remove** to modify the list of libraries or the name of an existing library.
4. Next, select the **ControlM** tab to enter the Control-M connection details.

   Note: As a prerequisite it is necessary to have installed the **Control-M Enterprise Manager** product.



Fill the available fields with the required data and then click **Add**. The Control-M folders are displayed in the central part of the screen.

- **ControlM Endpoint** - expects the entire link to the **Control-M REST API**. It includes the Control-M hostname, port, and the *automation-api* endpoint.

  ```
  Example: https://CMserverName:8443/automation-api
  ```

  Note: For a successful connection to the **ControlM Endpoint**, it is necessary to import the *rootCA.crt* certificate from **Control-M Enterprise Manager** in **Trusted Root Certification Authorities** on the same machine where **IBM AD Build Configuration** is installed.

  a. Go to **Start** > **Run** > **certlm.msc**.

  b. Right-click on the **Trusted Root Certification Authorities**, go to **All Tasks** and select the **Import...** button.

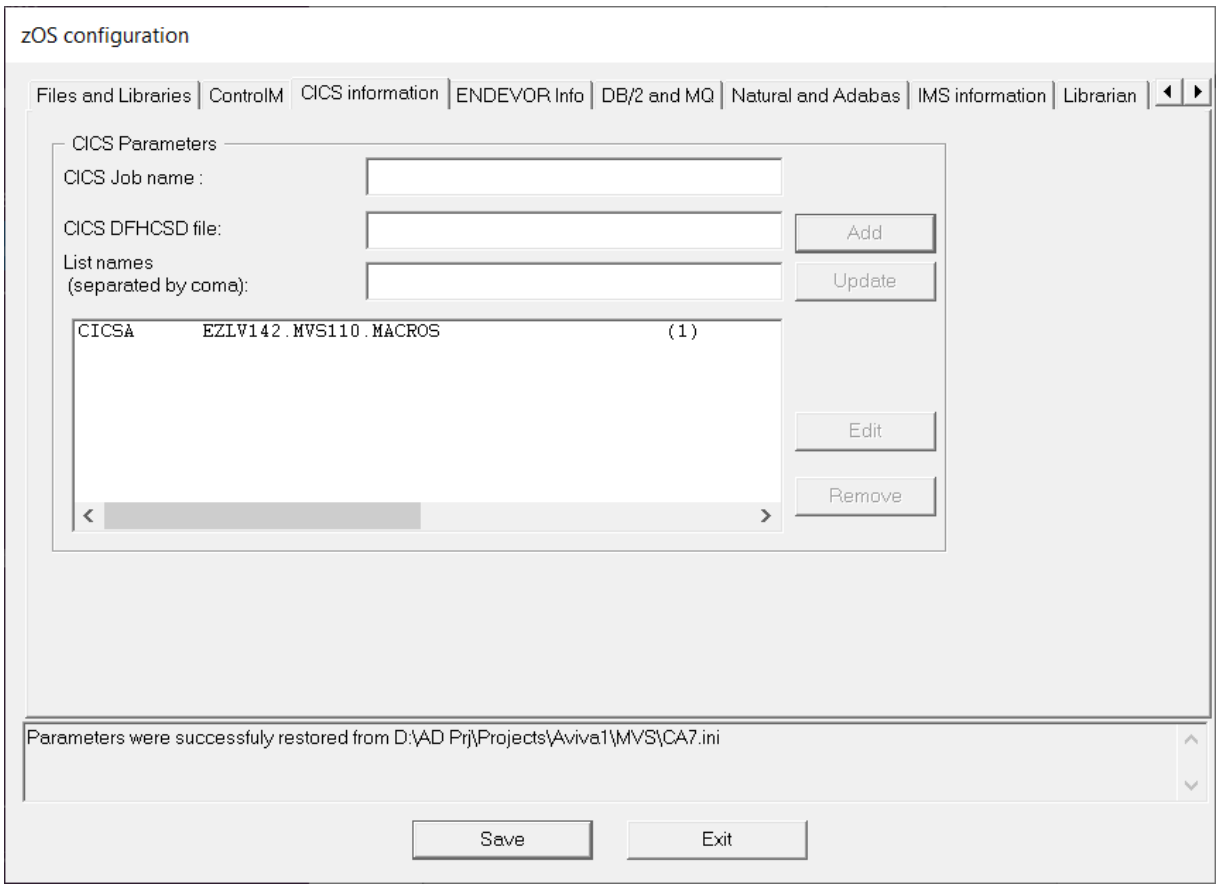
  c. Follow the steps that are present in the **Certificate Import Wizard**.
- **User name** - expects the username that is used to login in **Control-M Enterprise Manager**.
- **Password** - expects the password of the username that is used to access **Control-M Enterprise Manager**.
- **ControlM instance** - expects the name of the instance for which you want to retrieve the jobs.
- **ControlM folder** - expects the name of the folder for which you want to retrieve the jobs.

  Note: If you don't add any **ControlM folder** all jobs are retrieved from the mentioned instance.

In the **ControlM** tab the following actions can be performed:

- To edit the name of a Control-M folder, select it and then click **Edit**. Make the changes then click **Update**.
- To remove a Control-M folder from the list, select it then click **Remove**.

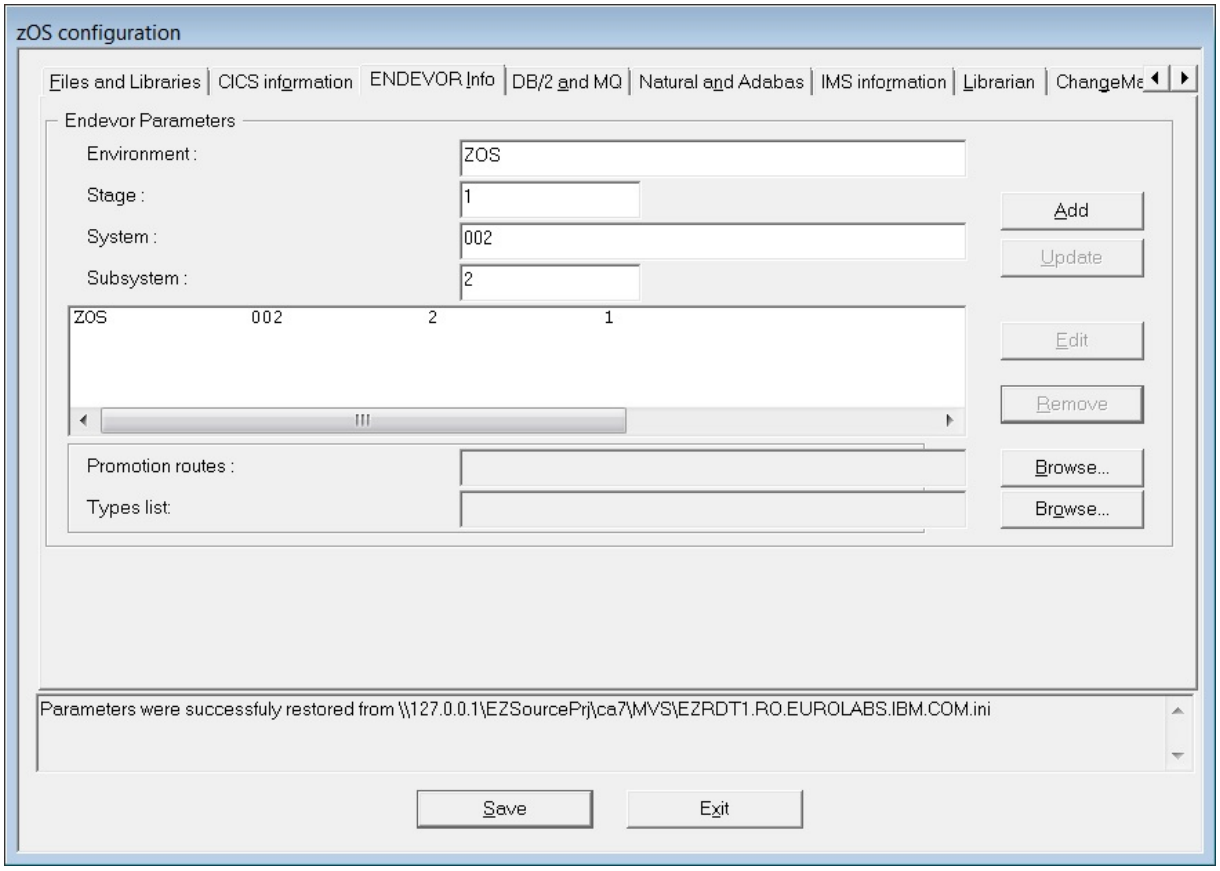
5. Next, select the **CICS information** tab to enter the CICS information library details.



Note: Only one CICS information library can be listed.

- To add a *CICS information library*, enter the details in the corresponding fields then click **Add**. The new library is displayed in the central part of the screen.
- To edit the details of the existing library, select it from the list and click **Edit**. The corresponding details are displayed. Make the changes then click **Update**.
- To remove the library, select it then click **Remove**.

6. Next, select the **ENDEVOR Info** tab to display it as in the following image:



Fill the available fields with the required data then click **Add**. The parameters file is added to the list in the central part of the tab.

- To edit the details of an entry, select it and then click **Edit**. The corresponding details are displayed. Make the changes then click **Update**.
- To remove a parameters file from the list, select it then click **Remove**.

Configurations for the **ENDEVOR Info** tab:

- **Endevor Parameters**

  Each entry in the list has four parameters. These parameters represent a state for which **IBM AD** collects sources. Whenever **IBM AD** looks for a member on Endevor for an entry in the list, the member is searched in the starting location according to these four parameters, and if not found, it continues to move forward on the promotion route by looking for the member until no forward movement can be done on the route. The four parameters are:

  – **Environment**

    Represents the functional area within an organization, for example: development, test, or production. There can be as many environments as needed within Endevor.

  – **Stage**

    Represents the stage of the software lifecycle and it is always associated with an environment that can have either 1 or 2 stages. Each stage has an ID that identifies whether it is the first or the second stage in the environment. It means that the ID is either 1 or 2 and it is unique only within the related environment.

  – **System**

    The user must define a system to each environment in which it is planned to be used. An additional usage is to define more than one system name to a mainframe application. It means that two systems can describe the same application in different environments and stages.

– **Subsystem**

  This is a subcomponent of a system. A system must have a minimum of one subsystem. The subsystem name for a mainframe application or component can be changed across environments and stages.

- **Promotion routes**

  The **Promotion routes** can be defined by using a configuration file. The **Promotion routes** file is optional, and if omitted then **IBM AD** does not trace the members forward on the promotion routes. A route represents a collection of maps, where a map is an indication of the current Environment + Stage + System + Subsystem, and the next Environment + Stage + System + Subsystem.

  The format of the configuration file is:

```
[<From Environment>,<From Stage>,<From System>,<From Subsystem>]:[<To Environment>,<To
Stage>,<To System>,<To Subsystem>]
```

  Example of the configuration file:

```
[DEV1,UNIT,SYS1,SUBSYS1]:[DEV1,INT,SYS1,SUBSYS1]
[DEV1,INT,SYS1,SUBSYS1]:[QA,QA,SYS1,SUBSYS1]
[QA,QA,SYS1,SUBSYS1]:[QA,HOLD,SYS1,SUBSYS1]
[QA,HOLD,SYS1,SUBSYS1]:[PROD,PROD,SYS1,SUBSYS1]
[DEV2,UNIT,SYS1,SUBSYS1]:[DEV2,INT,SYS1,SUBSYS1]
[DEV2,INT,SYS1,SUBSYS1]:[QA,QA,SYS1,SUBSYS1]
[DEV3,UNIT,SYS1,SUBSYS1]:[DEV3,INT,SYS1,SUBSYS1]
[DEV3,INT,SYS1,SUBSYS1]:[QA,QA,SYS1,SUBSYS1]
[QFIX,TSTFIX,SYS1,SUBSYS1]:[QFIX,PREPROD,SYS1,SUBSYS1]
[QFIX,PREPROD,SYS1,SUBSYS1]:[PROD,PROD,SYS1,SUBSYS1]
```

  From the **Promotion routes** field, click **Browse** and select the <PromotionRoutes> configuration file that contains the promotion routes. Make sure that you have access to the location where the <PromotionRoutes> configuration file is stored.

- **Types list**

  The **Types list** points to the file from which **IBM AD** reads all required types for the project. These types are categories of source code (COBOL, COPYBOOK, JCL, and so on). Types are defined to each system/stage in which the user plans to use them.

  The format of the types list file is:

```
Type1,Type2,Type3, and so on
```

  Examples of the types list file:

```
COBOL,JCL,COPYBOOK
```

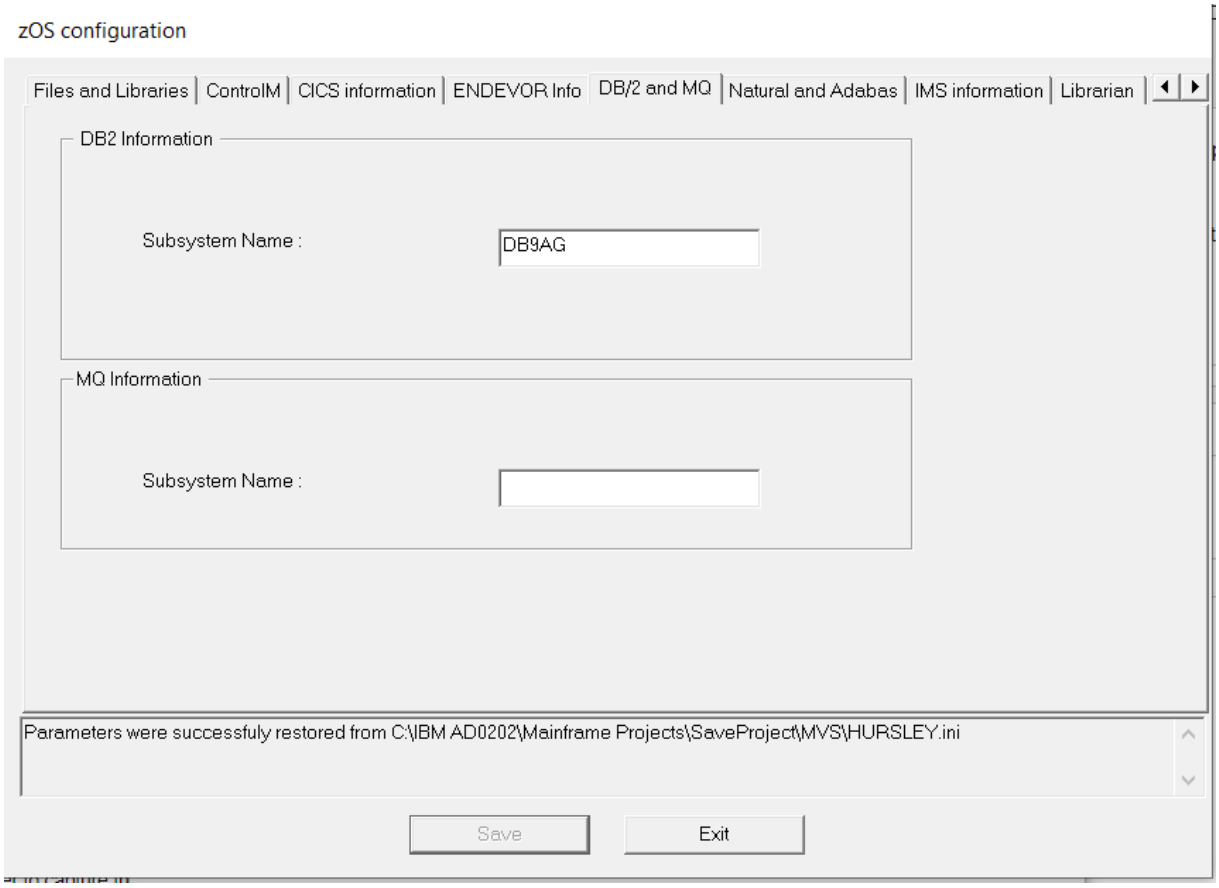
```
MACRO,JCL,ASSEM,PL1
```

  From the **Types list** field, click **Browse** and select the <TypesList> file that contains all Endevor supported file types that can be imported into the project. Make sure that you have access to the location where the <TypesList> file is stored.

  A sample file is available at the following location: <IBM AD Build Client installation folder>\Bin\Release\Samples\Endevor_Download\EndevorTypesList.txt.

7. Next, select the **DB2** and **MQ** tab to display it as in the following image:

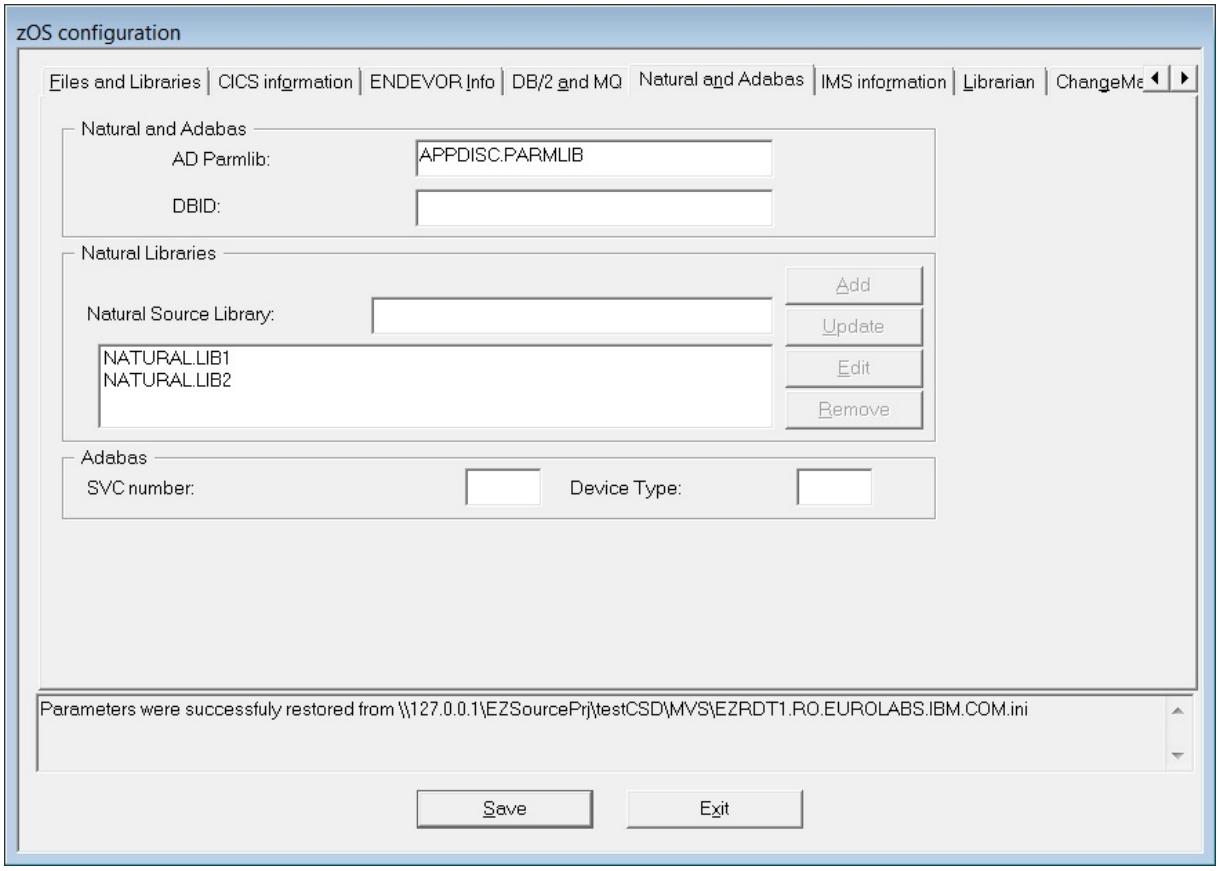
   Important: The minimum supported version for Db2 for z/OS is **11.1**.



Enter the corresponding **DB2** and **MQ** subsystem names in the associated fields.

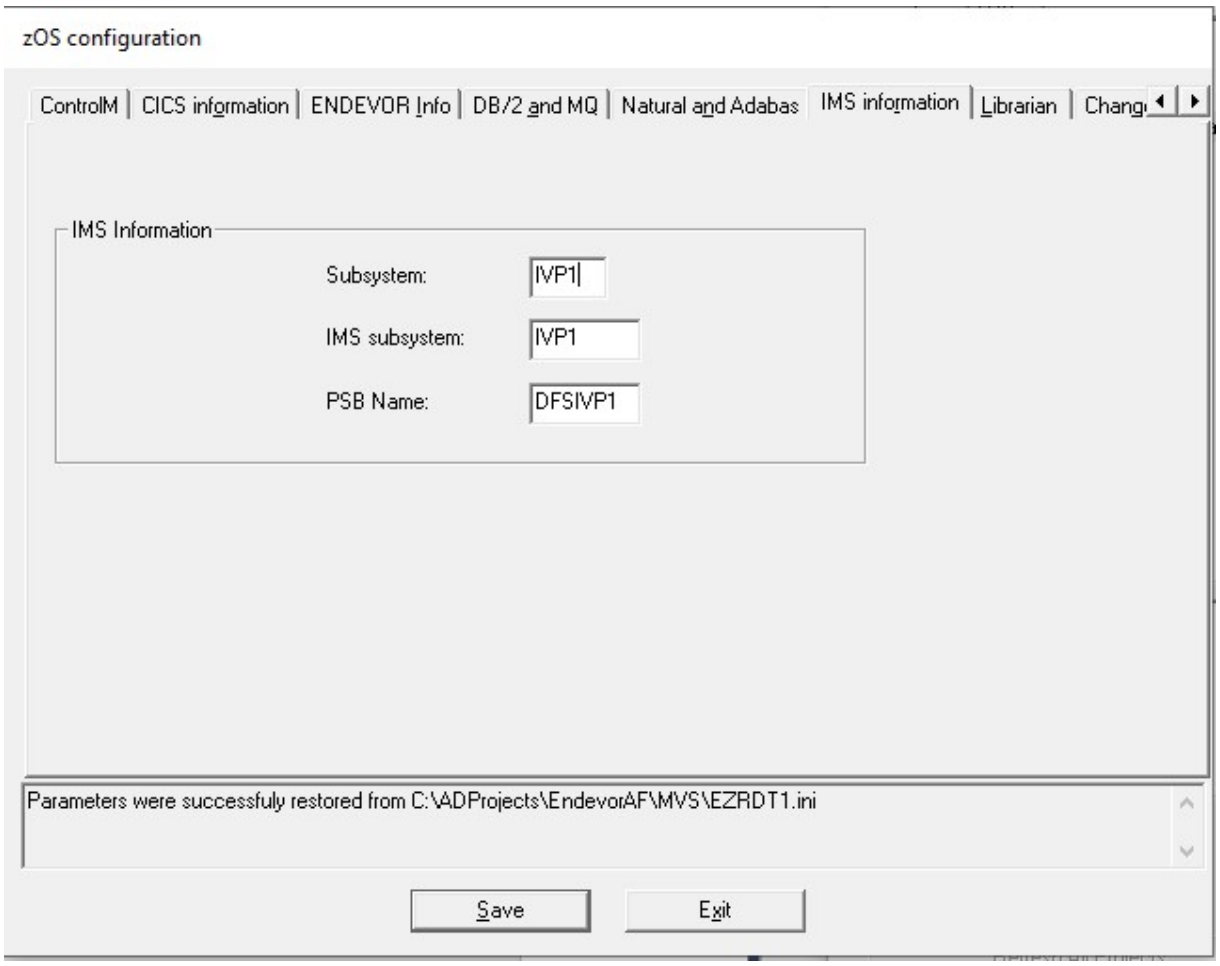
When configuring **DB2** on IBM AD Build Client, there are associated configurations that must be performed on the mainframe. For more information, see Db2 Checklist in Appendix 5.

8. Next, select the **Natural** and **Adabas** tab to display it as in the following image.



Complete the required settings information for the **Natural** and **Adabas** libraries. Use the prior procedure to add, edit, update, and remove libraries.

9. Next, select the **IMS Information** tab to display it as in the following image:



Important: IBM AD Connect for Mainframe retrieves IMS operational information by using the IMS Open Database Access (ODBA) callable interface. To make a connection to IMS, IMS ODBA must be configured to use resource recovery services (RRS=Y).

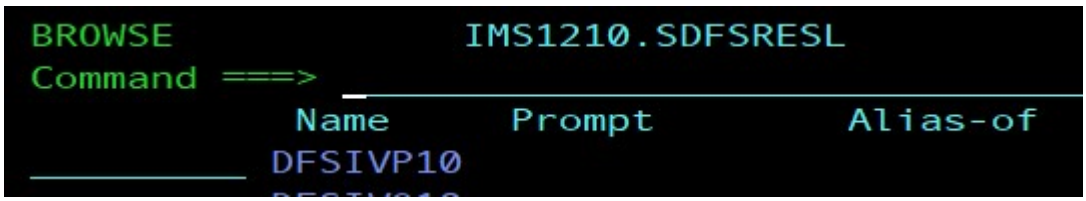
IMS ODBA is a separate product from IMS Open Database Manager (ODBM), although the two products might co-exist in an IMS environment.

Complete the fields to retrieve IMS information:

Subsystem
The four-character name of the IMS subsystem, for example, IVP1. This name can be obtained from the job log of the IMS control region in SDSF on the mainframe. If you do not know the name of the IMS control region, ask your IMS system programmer. The name is also printed at the far right of every line in the IMS started task job log.

IMS subsystem
The fourth to seventh characters of the IMS ODBA startup table name. The ODBA startup table resides in the IMS RESLIB and is usually named DFS*xxxx*0, where *xxxx* is the IMS subsystem name from the previous **Subsystem** field. However, double check with your IMS system programmer because the table does not have to follow this convention. *xxxx* might be set to something else at your site. Whatever the name of the table, when configuring IBM AD, enter only the four corresponding characters represented here by *xxxx*.

For this example, the ODBA startup table name is DFSIVP10, and *xxxx* is identical to the subsystem name entered in the **Subsystem** field: IVP1.



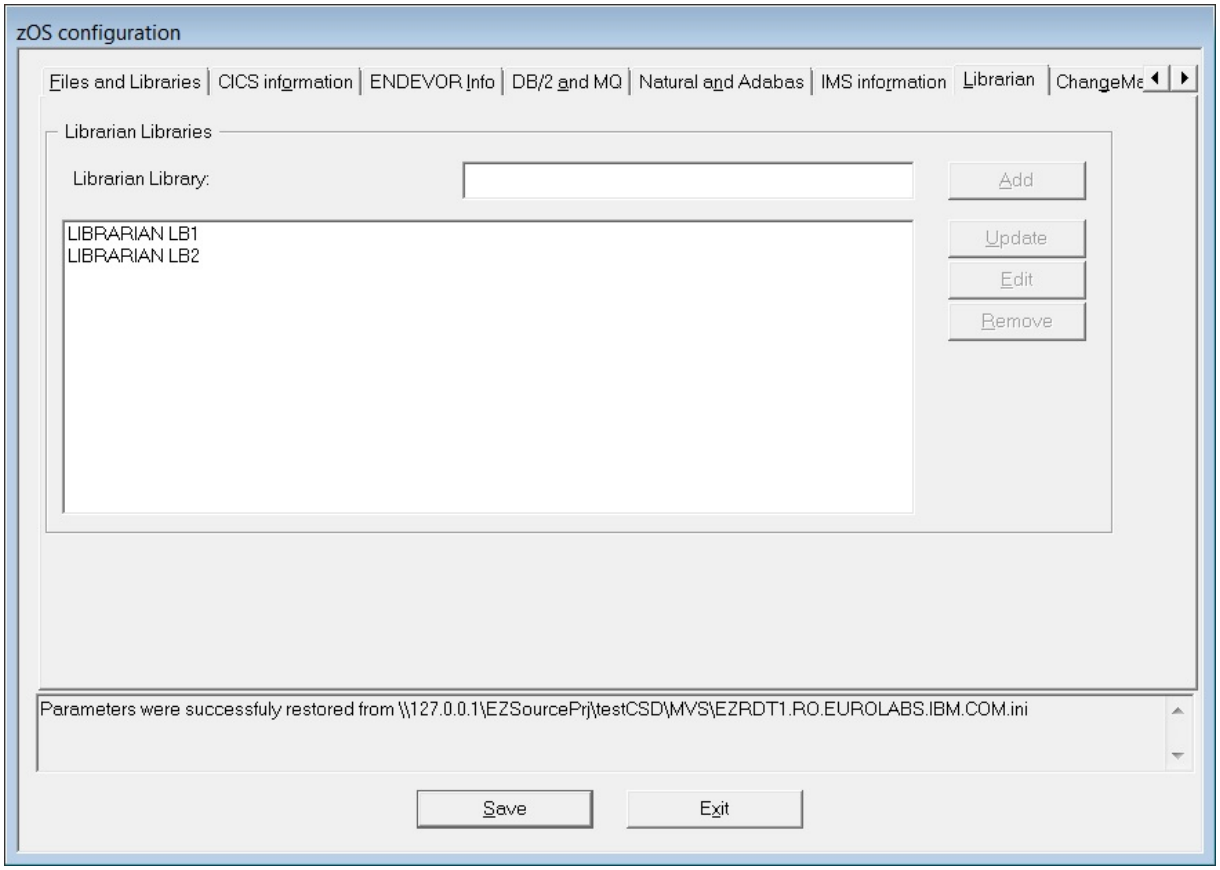
For more information about the ODBA startup table, go to the IMS documentation.

PSB Name

The PSB name. The PSB name is application-dependent, so knowledge of the application that is analyzed is needed. If the application uses multiple PSBs, select only one for this field.

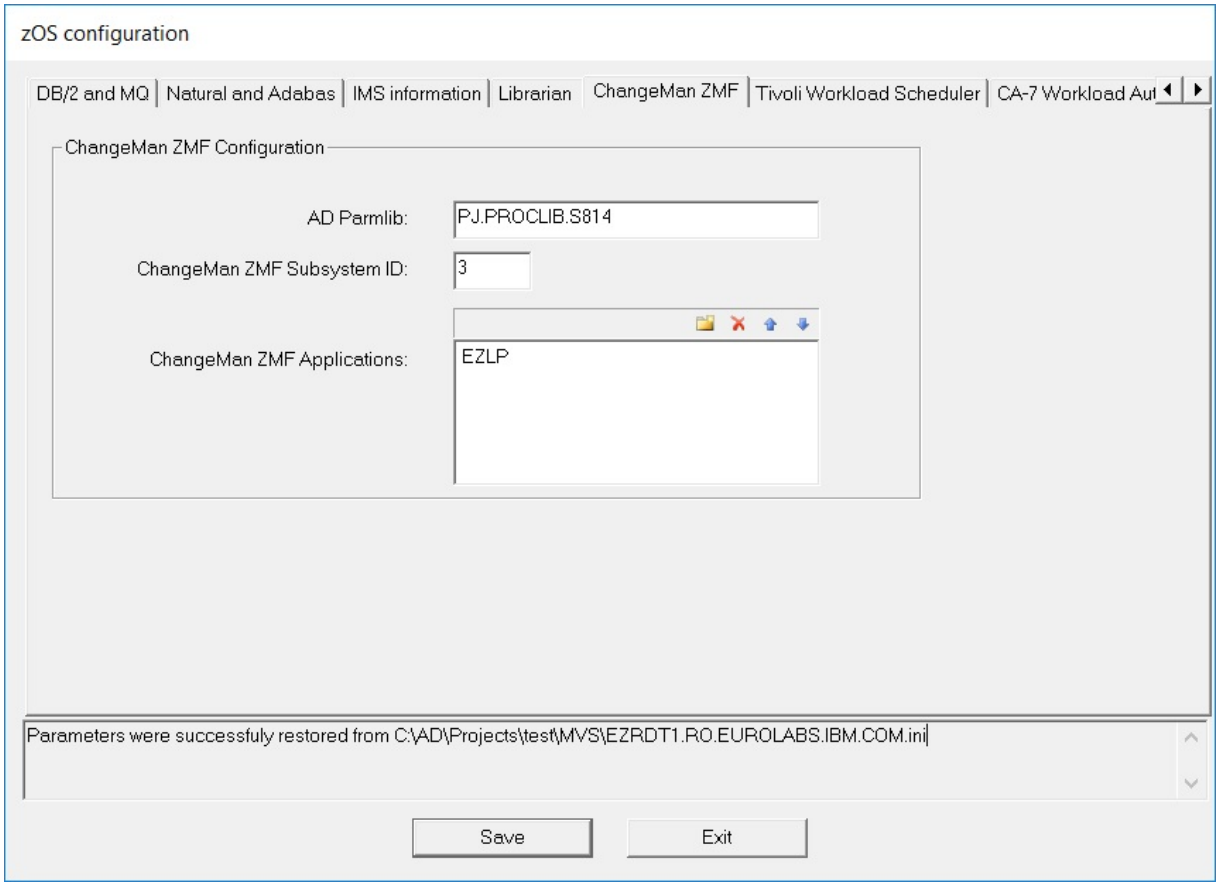
You can set this field to any PSB name that exists on the IMS system, and then a complete list of all the IMS databases, transactions, and program names that are defined to this particular IMS system can be retrieved.

10. Select the **Librarian** tab to display it as in the following image:



Add the names of the required libraries and use **Add**, **Edit**, **Update**, and **Remove** to manage these libraries.

11. Select the **ChangeMan ZMF** tab to display it as in the following image:



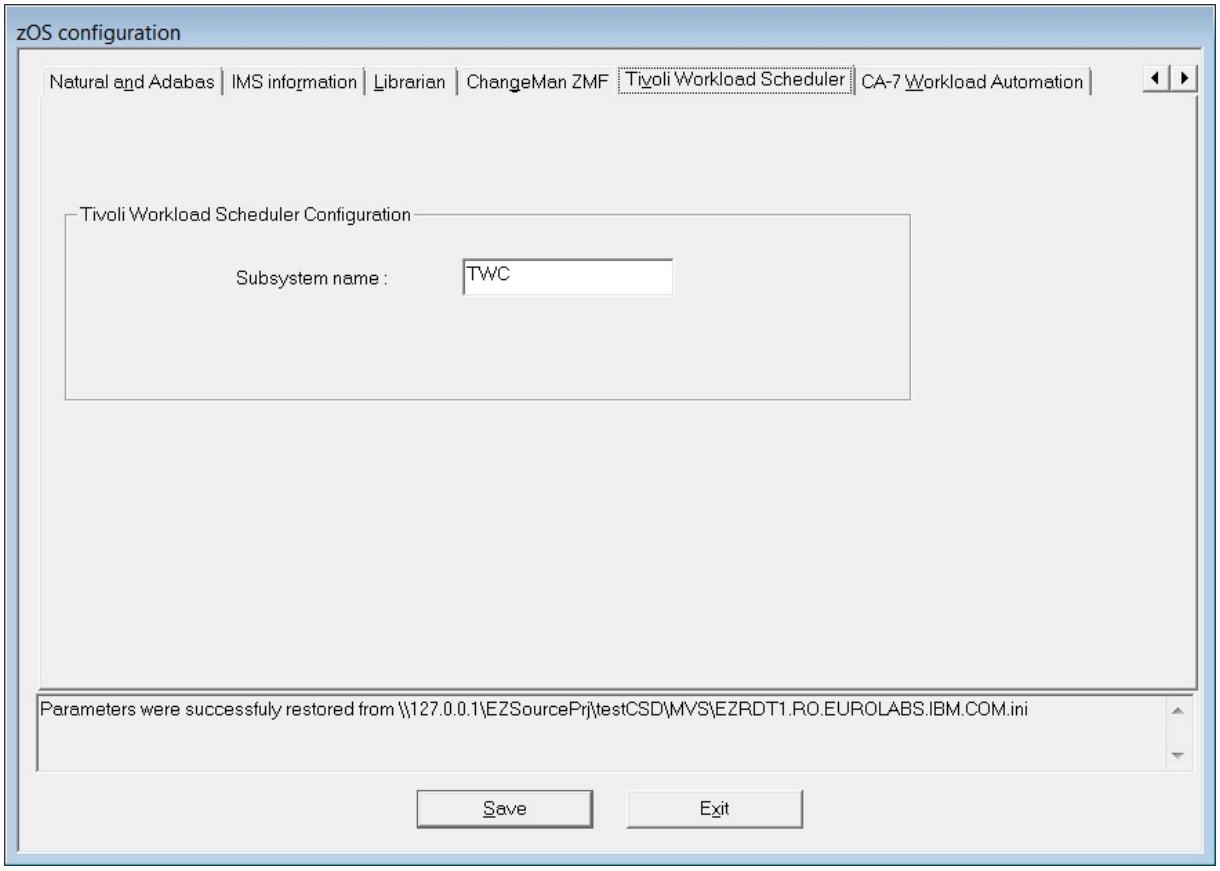
Complete the name of the IBM ParmLib library and then enter the number of the **ChangeMan ZMF Subsystem**.

In the **ChangeMan ZMF Applications** field, click  to add an application name to the list. Click  to remove the selected application from the list. Use  and  to change the position of the selected application in the applications list.

Note: If no **ChangeMan ZMF Applications** are mentioned, the members are scanned/retrieved only from packages.

When configuring **ChangeMan ZMF** on IBM AD Build Client, there are associated configurations that must be performed on the mainframe. For more information, see ChangeMan® ZMF Checklist in Appendix 5.

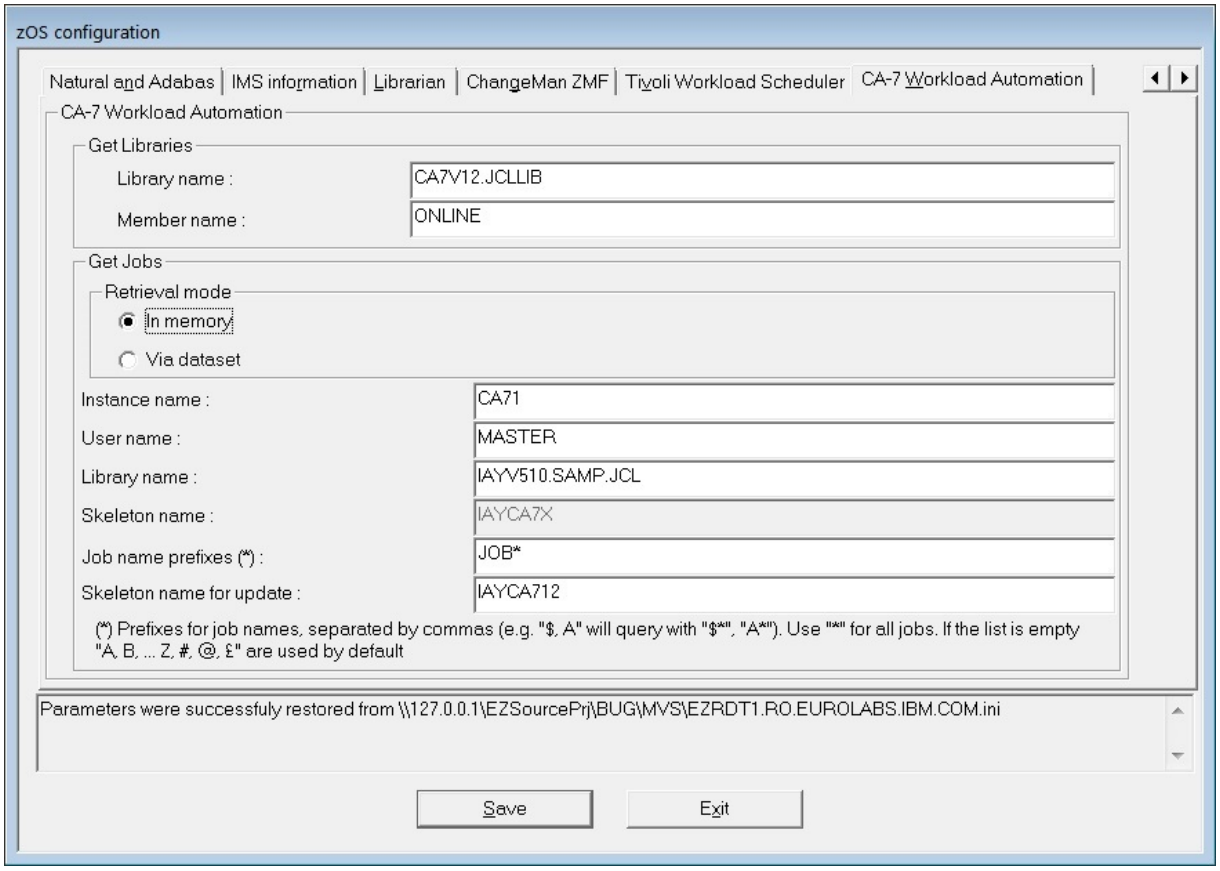
12. Select the **Tivoli Workload Scheduler** tab to display it as in the following image:



Complete the Subsystem name of the TWS controller. After you finish, click **Save**.

13. Select the **CA-7 Workload Automation** tab to display it as in the following image. Select a retrieval mode, and specify values in the fields.

    If the settings in the CA-7 Workload Automation pane are changed, run a full import, instead of an update, to assure the consistency of the information in the repository with the one from the mainframe.



Library name
The name of the library where the INIT deck is located.

Member name
The name of the INIT deck member that lists all the libraries, in which CA-7 must look for to find and submit jobs.

In the CA-7 documentation, the INIT deck is sometimes referred to as the CA-7 Initialization file. To find this library (PDS) and member combination for your system, refer to the JCL for the CA7ONL procedure at your site. It is referenced in this JCL by DD name UCC7IN.

Note: If you do not have access to this JCL, ask your CA-7 administrator for the information.

In memory
The default retrieval mode. It uses CA Common Communications Interface (CAICCI) to get job information. The results are retrieved directly from CA-7 into the memory address space of the agents and transferred back to IBM AD Build.

Via dataset
The alternative retrieval mode. It uses Batch Terminal Interface (BTI) to get job information. The results are delivered by CA-7 into a data set. Then the agents read the data and transfer it back to IBM AD Build.

Instance name
The CA-7 instance name.

User name
The name of the user under which the agents are running.

Library name
The name of the library that contains the AD skeleton. This library is a member that is delivered as part of the installation if the CA-7 access through BTI is required.

Skeleton name
The name of the AD skeleton that is used for accessing CA-7 through BTI. This skeleton is submitted by the agents. The job uses the BTI interface to get the information back to the data set.

Note:

- This field is available only if the **Via dataset** retrieve mode is selected.
- The value in this field is used when **Full CA-7 import** is selected when you retrieve operational information.

Job name prefixes
The name prefixes of the jobs to be retrieved. Separate multiple job name prefixes by commas. If this value is not specified, all the jobs with a name that starts with any of the following characters are to be retrieved:

A-Z # @ £

Note: This field is available only if the **In memory** retrieve mode is selected.

Skeleton name for update
The name of the AD skeleton that is used for accessing CA-7 through BTI. This skeleton is submitted by the agents. The job uses the BTI interface to get the information back to the data set.

Note: The value in this field is used when **Update CA-7 information with the differences since the last import** is selected when you retrieve operational information.

Bringing Operational Information

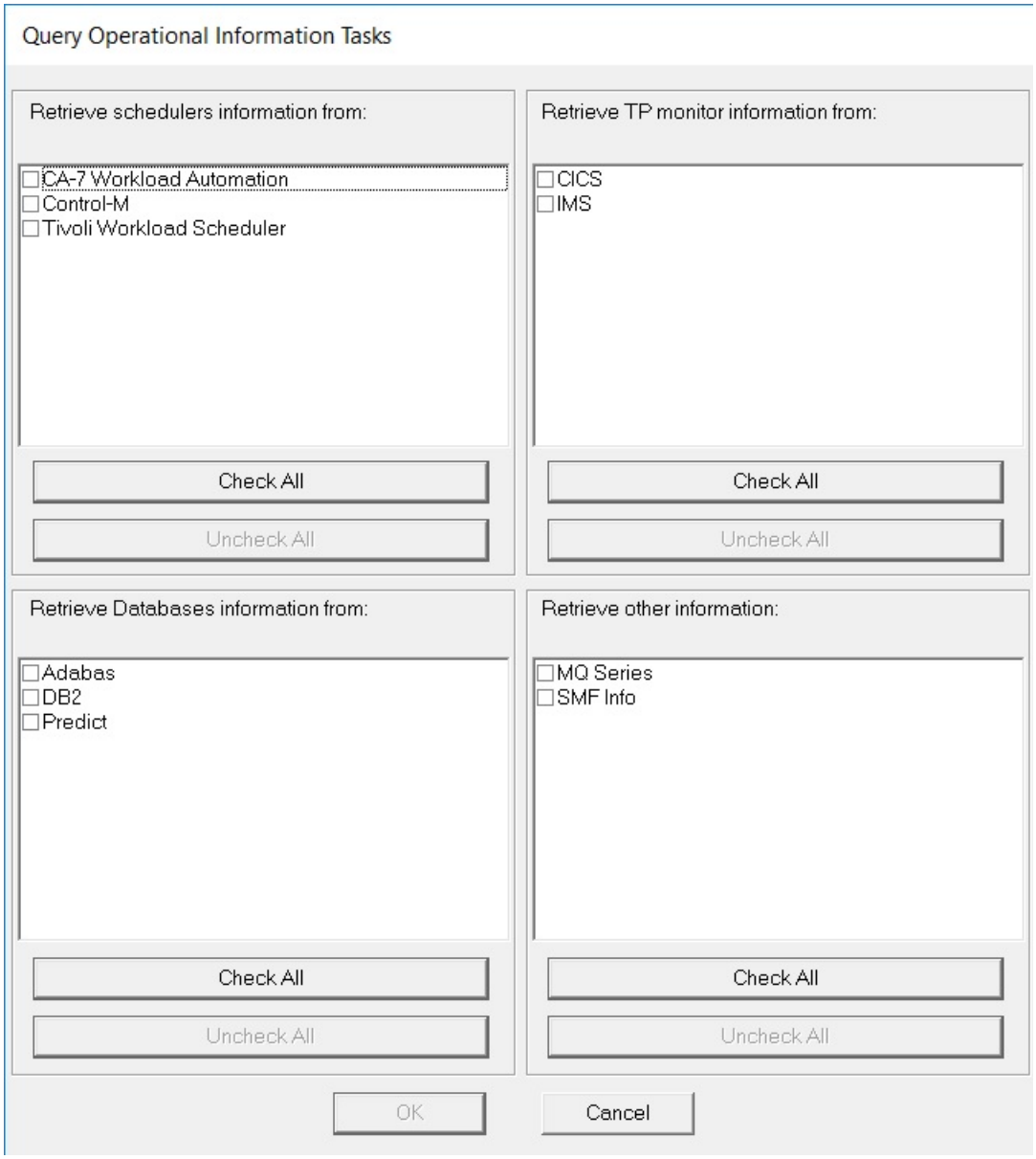
To retrieve operational information, follow these steps:

1. Configure the z/OS connection (for details see “Configuring the z/OS Connection” on page 216)
2. Retrieve operational information.

Retrieve Operational Information

Note: Before you retrieve operational information, you must configure the z/OS connection. If you associate the z/OS connection to a project only but you do not configure it, you cannot retrieve the data. For more information, see topic “Configuring the z/OS Connection” on page 216.

Go to **IBM Application Discovery Build Configuration**, select the **Projects** tab, select the project, select the z/OS node, right-click to display the menu, and select the **Retrieve Operational Information** option to display the following window.



Note: If you do not configure the z/OS connection for the selected task, a warning message is presented, indicating the entries for which the configuration is missing.

A list of available tasks is presented grouped by categories: You can choose to **Retrieve schedulers information**, **Retrieve TP monitor information**, **Retrieve Database information**, or **Retrieve other information**. Make the selections then click **OK** to run **Query Operational Information**.

Note: The available options depend on the type of project you create.

Control-M

If the **Control-M** task is selected, it brings information about the Control-M jobs for the selected project based on the **z/OS Connection** configuration. For more information, see Configuring Control-M tab. The retrieved information is in XML format and it is automatically added by the task in the project's `ControlM` folder.

Considering that you can attach more than one **z/OS Connection** to a project, an XML file is retrieved for each connection.

After the task is successfully executed, for a proper Control-M analysis, the IBM AD project needs to be built. Instructions for performing a Build of the project can be found at “Building Projects” on page 264.

Following is a list of entities/objects for which **Connect for mainframe** brings information:

- Adabas

  The **Connect for mainframe** gives information about a total of three entities:

  1. Adabas Database.
  2. Adabas File.
  3. Adabas Field.

  The information refers to physical allocations, defined files that include file fields.

- CA Workload Automation CA 7 Edition

  The **Connect for mainframe** gives information about a total of two entities:

  1. CA 7 Jobs.
  2. Datasets.

  The information refers to triggering and dependencies.

  The following two options are provided for CA-7 Workload Automation:

  Full CA-7 import
  Removes the existing information that is stored in the repository, and then imports the information from the mainframe into the repository as configured.

  Update CA-7 information with the differences since the last import
  Import only the changes from the mainframe into the repository as configured.

  Note: A full import must be run before an update. Otherwise, the update will not have the correct results.

  For more information about the CA-7 configuration in AD Connect for Mainframe, see "Configuration for CA-7 " on page 155.

- CSD (CICS)

  The **Connect for mainframe** gives information about a total of seven entities:

  1. CICS Region
  2. CICS Group
  3. CICS List
  4. CICS File
  5. CICS Map
  6. CICS Program
  7. CICS Transaction

  The information refers to all CICS regions, including defined programs\transactions\maps\files, hierarchy of groups and lists, and performance information for transactions such as **Elapsed time**, **IO count**, and **DB2 count**.

- DB/2

  Connect for mainframe brings information about a total of 14 entities:

  1. Db2 Database.
  2. Db2 Table.
  3. Db2 Column.
  4. Db2 Index.
  5. Db2 Key.
  6. Db2 Package.
  7. Db2 Package List.

8. Db2 Plan.
9. Db2 Storage Group.
10. Db2 Stored Procedure.
11. Db2 Table Space.
12. Db2 Trigger.
13. Db2 View.
14. Db2 Volume.

The information refers to all databases, tables, columns, table spaces, triggers, views, plans storage groups.

The operational information data about Db2 is only stored in the repository and it is not downloaded on the disk. The Mainframe Layout graph, from IBM AD Analyze Client, is showing basic details coming from Db2.

- Devices and Physical files

  The **Connect for mainframe** gives information about the following entities:

  1. Physical files
  2. Devices

  The information refers to all devices connected to the LPAR, and all physical data sets on disk devices.

- Hardware

  The **Connect for mainframe** gives information about one entity:

  1. CPU

- IMS

  The **Connect for mainframe** gives information about a total of three entities:

  1. IMS Database.
  2. IMS Transaction.
  3. IMS Program.

  The information refers to databases, programs, and transactions of IMS.

- LPAR and SYSPLEX

  The **Connect for mainframe** gives information about a total of two entities:

  1. LPAR Information.
  2. Sysplex Information.

  The information refers to LPARS, SYSPLEXS, the relationships, and links to other entities (CPU, installed software)

- MQ Series

  The **Connect for mainframe** gives information about a total of three entities:

  1. Queue Manager
  2. Queue
  3. Channel

  The information refers to Queue Managers, Queues, and Channels.

- Predict

  The **Connect for mainframe** gives information about a total of two entities:

  1. Predict file.
  2. Predict field.

The information refers to predict files and fields.

- SMF

  The **Connect for Mainframe** gives information about a total of three entities:

  1. SMF JCL information.
  2. SMF JCL Step information.
  3. SMF JCL Step IO.

  The information refers to scheduled jobs, including total CPU consumption, IO count, and CPU count per steps, and used physical files.

- Tivoli Workload Scheduler

  The **Connect for mainframe** gives information about a total of two entities:

  1. TWS Applications.
  2. TWS Jobs.

  The information refers to TWS defined applications, jobs, and their dependencies.

  Before you retrieve the information, a confirmation window is displayed. Click **Yes** to start the selected query operation.

  After the operation is finished, 2 log files are generated: One containing the errors (if applicable), the other detailing the operations undertaken. These log files are stored in the following location: `project name/z0S/logs/z0S name`.

Retrieve operational information in the background

IBM AD Build Client provides an option to retrieve operational information in the background. To force the retrieve operation information to run in the background, follow the steps:

1. Click **Start**, select **Run** then type **cmd** followed by **ENTER** to open the command window.
2. Go to the folder where your IBM AD Build Client is installed and locate `IBMApplicationDiscoveryBuildConfiguration.exe`.
3. Drag `IBMApplicationDiscoveryBuildConfiguration.exe` file into the command window then type **/?** and press **ENTER**. A window is displayed containing detailed instructions about how to retrieve operational information in the background. In the command prompt enter the following command:

   ```
   C:\Program Files\IBM Application Discovery Build
   Client\Bin\Release\IBMApplicationDiscoveryBuildConfiguration.exe /ba <fully qualified file
   name>
   ```

   The *<fully qualified file name>* points to an INI file that contains information about actions that are executed in headless mode.

   A sample for the configuration INI file is found in `C:\Program Files\IBM Application Discovery Build Client\Bin\Release\Samples\BuildConfigurationBASample.ini`. You can use the sample INI file or create a new configuration file.

   The format of *<fully qualified file name>* is as follows:

   ```
   [OperationalInformation]
   ;CA7IMPORT=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;CA7UPDATE=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;TWS=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;CSD=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;IMS=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;Adabas=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;DB2=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;Predict=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;SMF=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;MQ=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]...
   ;CONTROL_M=[<ProjectName1>, <zOSConnectionName1>],[<ProjectName2>, <zOSConnectionName2>]
   ```

Attention: In order for the operational information to be taken into account, the comment tag **;** must be removed.

Predefined values for *<Operational task>*:

- Schedulers information: CA7IMPORT, CA7UPDATE, TWS
- TP monitor information: CSD, IMS
- Databases information: ADABAS, DB2, Predict
- Other information: MQ, SMF

4. A log file is generated in `C:\Program Files\IBM Application Discovery Build Client\Bin\Release\Log` folder. The name of the log file is `ADBuildConfiguration_datetime.log`.

Bringing data from mainframe libraries (PDS Libraries, Endevor, Librarian, Natural)

1. Associate a z/OS connection to project (for details see Associating a z/OS Access Point to a Project).
2. Configure the z/OS connection (for details see “Configuring the z/OS Connection” on page 216).
3. Query the environment.
4. Scan files.
5. Scan libraries.

Following is the detailed description of the steps 3-5.

Note: IBM AD Connect for Mainframe uses z/OS Unicode services to convert character data from one code page to another. There are two settings that tell IBM AD Connect for Mainframe which code page to use for the host and the client:

1. HOST CODE PAGE
2. CLIENT CODE PAGE

Each variable is a five digit number, denoted by a Coded Character Set Identifier (CCSID) and established by the Character Data Representation Architecture (CDRA). Usually the CCSID is the same as the code page number used in the informal use. In case there is a doubt in using the correct CCSID, see z/OS Unicode Services User’s Guide and Reference, Appendix A. Description of CCSIDs.

Retrieving Source Code Information

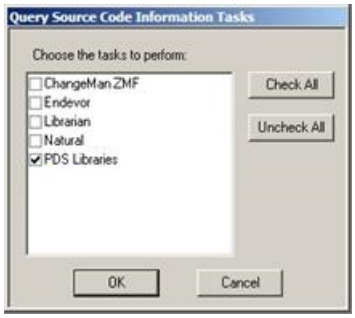
Warning: Before you retrieve the source code information, you must configure the z/OS connection. If you associate the z/OS connection to a project only but you do not configure it, then you cannot retrieve the data.

This step reads the contents of the sources (either auto-discovered or manually added) configured as presented in “Configuring the z/OS Connection” on page 216.

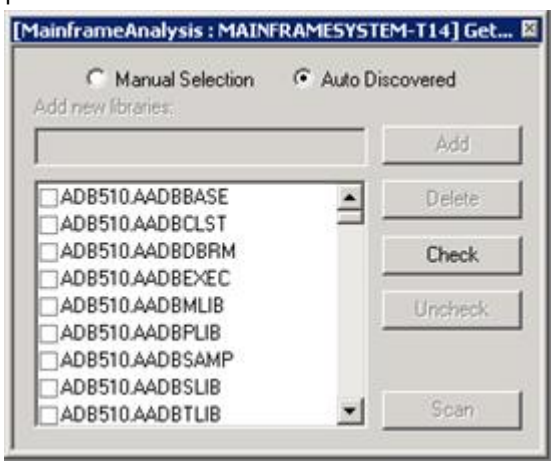
Goto **IBM Application Discovery Build Configuration**, select **Projects** tab, then select the project, then select the z/OS node, and then right-click to display the menu.

Note: You can run this operation on several projects simultaneously through several z/OS connections. Select all the z/OS connections that you want to use (several connections that are defined for the same project or for different projects) then right-click to display the menu and select the scanning operation that you want to apply. If the scan operation cannot be completed on one or more of the selected z/OS connections, a warning message is displayed but the scanning operation continues for the rest of the selected connections. Also, for each z/OS connection a dialog box offers the possibility of individually selecting the libraries to be scanned.

1. Select the **Retrieve Source Code Information** option to display the following window:



If libraries are found on the remote computer, a list of libraries that are automatically discovered are presented.



To scan the libraries added manually, select the **Manual Selection** option. The list is updated to present only those libraries. Select the libraries that you want to include in the scanning process, then press **Scan**. Alternatively, you can manually add a library to the list.

Note: The **Delete** option becomes available if you select a manually added library.

2. If the **PDS Library** option is selected,
   Connect for mainframe brings information about a total of three entities:

   a. Source Library

   b. Source Member

   c. User

   The information refers to libraries and members within the libraries, including member creation date, modification date, and users who modified members.

3. If the **Endevor** option is selected, the scanning window displays a list of libraries that are selected for the scanning operation.
   Connect for mainframe brings information about the following CA Endevor entities:

   a. Endevor members.

   b. Endevor Systems.

   c. Endevor Subsystems.

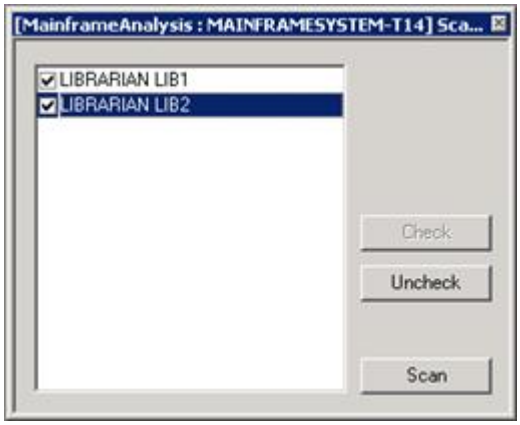
   d. Endevor Environments.

   e. Endevor Stages.

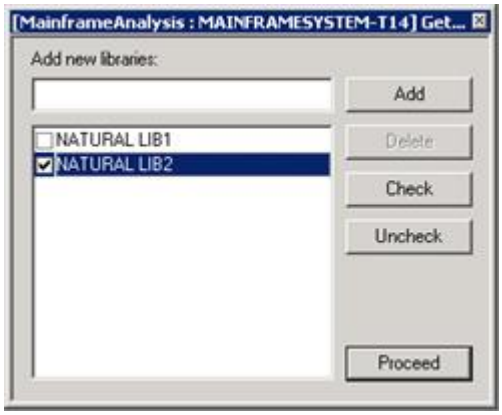
   f. Endevor Types.

   The information refers to System, Subsystem, Environment, Stage, and type.

4. If the **Scan Librarian Libraries** option is selected, the scanning window is displayed.



5. If the **Scan Natural Libraries** option is selected, the scanning window is displayed.



From the list, select the Natural libraries that you want to scan, then click **Proceed**. Alternatively, you can use the **Add new libraries** field to add new libraries to the list of existing ones.

Connect for mainframe brings information about the following Natural entities:

a. Natural Library

b. Natural Member

The information refers to last update date, version, and user name.

6. Before scanning, a confirmation window is displayed: Click **Yes** to start scanning the selected libraries.

Note: After the operation is finished, 2 log files are generated: One containing the errors (if applicable) the other detailing the operations that are undertaken. These log files are stored in `project name/z0S/logs/z0S name`.

7. Repeat the procedure to scan all the libraries.

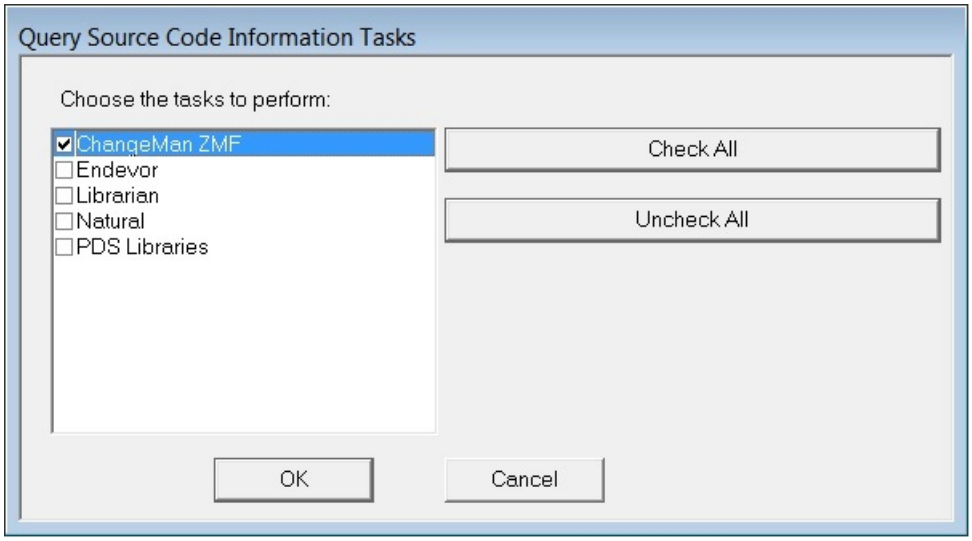
Bringing Data From Mainframe Using ChangeMan® ZMF

1. Associate a z/OS connection to project (for details see Associating a z/OS Access Point to a Project)
2. Configure the z/OS connection (for details see “Configuring the z/OS Connection” on page 216)
3. Retrieve ChangeMan information.

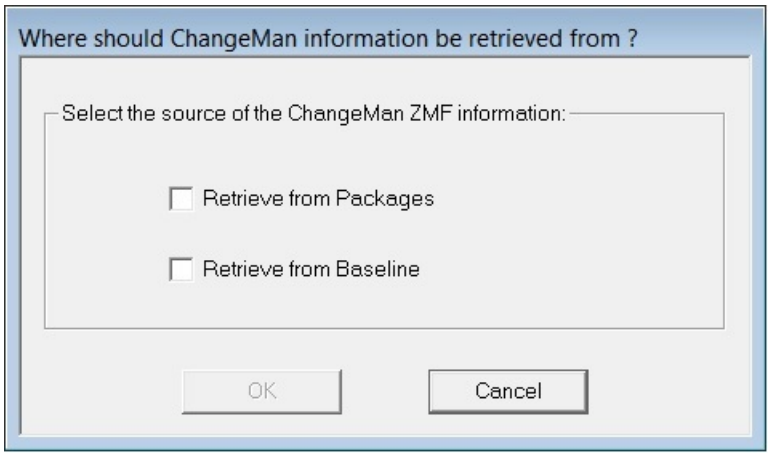
Following is the description of the step 3.

Retrieving ChangeMan® Information

1. In the **IBM Application Discovery Build Configuration**, select the project and then right-click to display the corresponding menu. Click **Retrieve source code information**.



2. If applications are configured in the **z/OS Configuration** > **ChangeMan ZMF**, then the next screen is displayed after you click **OK**.



Select **Retrieve from Packages** or **Retrieve from Baseline** or both.

ChangeMan ZMF

The **Connect for Mainframe** brings information about a total of four entities:

a. ZMF Applications.

b. ZMF Components.

c. ZMF Libraries.

d. ZMF Packages.

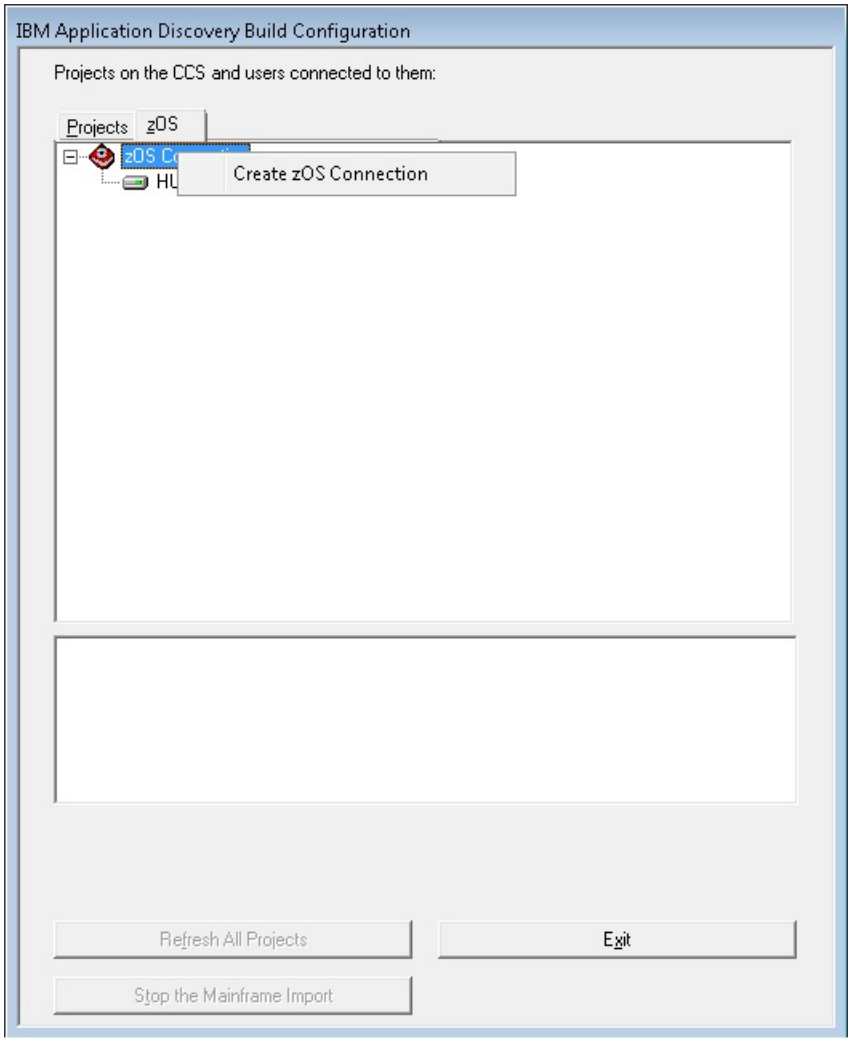
The information refers to Applications, Components, Libraries and Packages, including relationships such as Library to Component, Package to Component, Packages to Application.

If applications are not configured in the **z/OS Configuration** > **ChangeMan ZMF** (**ChangeMan ZMF** tab), a warning message is displayed after you click **OK**.

The zOS Tab

When you work on a project, you might want to use resources that are on a mainframe machine. You can set up your remote connections and associate them to your project, by using the **zOS** option.

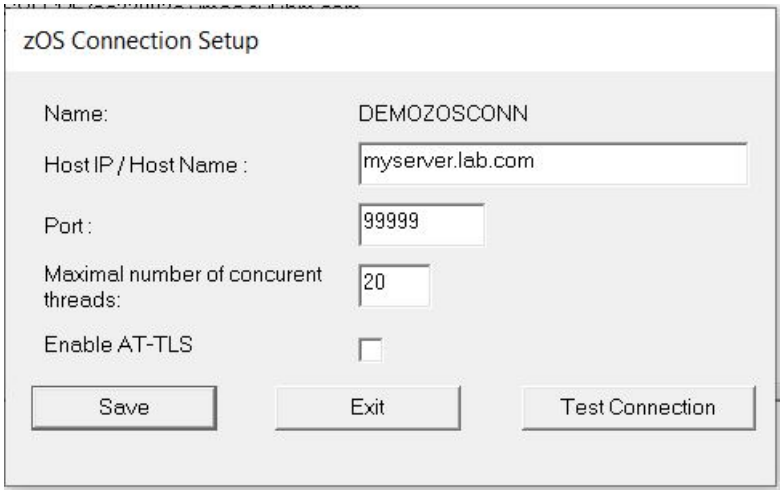
The **zOS** tab presents a list of all the z/OS connections that are defined and you can modify the existing ones or create new connections.



Create a new z/OS connection

To create a new z/OS connection, follow the steps:

1. Go to the **zOS** tab in **IBM AD Build Configuration**, right-click on the z/OS connection root and select **Create zOS Connection**. The **Mainframe Access Point Connection** window is displayed.
2. Enter the name for the new z/OS connection, then click **OK**.
3. The **zOS Connection Setup** dialog is displayed.



Enter the host IP or host name, the port number, and the maximal number of concurrent threads in the corresponding fields, then click **Test Connection** to check the connection to the new Mainframe access point. If the parameters you entered are correct, a confirmation window is displayed. Starting with version 5.1.0.7, the **Enable AT-TLS** checkbox is available. With this checkbox, encrypted and non-encrypted connections can be supported at the same time. To create an encrypted connection, select this checkbox and follow the instructions in topic "Enabling TLS Connection between IBM AD Build Client and IBM AD Connect for Mainframe" on page 291.

Note: An existing z/OS connection can be edited or deleted.

Automatic Messaging

When multiple users are connected to the same project and one user changes the folder structure of the project or attempts one of the following operations: **Delete project**, **Associate z/OS**, **Recreate repository**, **Upgrade repository**, **Restart Application Server**, all the other users receive an automatic notification.

Enabling the Assembler Macro Expansion Feature

The **Assembler Macro Expansion** feature is an extension of the base Assembler compiler and it is not enabled by default. To enable the **Assembler Macro Expansion** feature, follow these steps:

1. Right-click **IBM AD Build Configuration**, and select **Run as administrator**.
2. In the **IBM AD Build Configuration** window, select your **project**, and choose **Options**.
3. In the **Options** window, select **Enable Macro Expansion** and click **OK**.

As a result, the **Assembler Macro Expansion** feature is enabled for the selected **project**.

Enabling the z/OS User Authentication feature for IBM AD Build Configuration

Starting with **IBM AD V6.0.1** you can use your z/OS user ID and password credentials and based on your access privileges you can pull the source files from the z/OS source systems.

Before you perform any action, you must take into account the following observations:

- Make sure that the AT-TLS setup for IBM AD Connect for Mainframe is **completed** and the TLS connection between **IBM AD Build Client** and **IBM AD Connect for Mainframe** is **enabled**. For more information, see "Enabling TLS Connection to IBM AD Build" on page 183 and "Enabling TLS Connection between IBM AD Build Client and IBM AD Connect for Mainframe" on page 291.
- Make sure that you are using the latest version of **IBM AD Connect for Mainframe**. There is no compatibility with an earlier version of **IBM AD Connect for Mainframe**.
- The **z/OS User Authentication** feature is available only for the z/OS mainframe instances that are configured with TLS and enabled in the `z0S-Data.ini` configuration file. For more information, see "Configurations" on page 239.
- All operations are performed **only** in the **graphical user interface** (**GUI**) of **IBM AD Build Configuration**.
- The `Please enter the z0S User/Password in order to acces the z0S Access Point` message is prompted when performing mainframe-related operations in **IBM AD Build Configuration**.

z/OS Username and Password Guidelines

- For the **IBM AD V6.0.1** release, only a standard of 8 character mixed case password is supported. Password phrases are not supported.
- The username and password are volatile and they are kept in the memory for the running instances of **IBM AD Build Configuration**.
- If you close **IBM AD Build Configuration** and later you want to perform any mainframe-related operation it will be necessary to reenter the username and password. Closing the application is

considered a reset process for the z/OS credentials. It shall be used in case of wrong or modification of the z/OS credentials.

- In case that a project has multiple z/OS mainframe instances available, the username and password is requested for each of the z/OS mainframe instance that is used in the related-mainframe operation and only once per **IBM AD Build Configuration** instance.

  Note: If other projects have the same z/OS mainframe instance attached and the z/OS mainframe instance was already prompted in a project, it will not be subsequently prompted for other projects that are on the same z/OS mainframe instance.
- If a mainframe-related operation is started on a selection of multiple z/OS mainframe instances and if you cancel entering the username and password for one of the z/OS mainframe instances, the whole operation will be aborted.

Configurations

To enable z/OS User Authentication feature, perform the following configurations:

1. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\Samples` and copy the `zOS-Data.ini` configuration file in the `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release` folder.
2. Open the `zOS-Data.ini` configuration file by using a text editor and enter the desired values for the parameters that are detailed below. By default, the `zOS-Data.ini` configuration file has the following format:

   ```
   [MVS1]
   use_cred=Y/N

   [MVS2]
   use_cred=Y/N
   ```

   Where:

   - **[MVS1]** and **[MVS2]** are the names of the z/OS mainframe instances that are present in **IBM Application Discovery Build Configuration** > **zOS** Tab.

     Important: The square brackets need to be preserved and between them, the actual name of the z/OS mainframe instance needs to be added with no spaces nor tabs between the brackets.
   - **use_cred** is an attribute that expects the **Y** or **N** values. This attribute decides whether the mentioned z/OS mainframe instance is prompted for the username and password or not.

     Note: The attribute value is not case-sensitive.

   Important: In case that a z/OS mainframe instance is not present in the configuration file, or it is present but the attribute value is set to **N**, for the related z/OS mainframe instance, the `Please enter the zOS User/Password in order to acces the zOS Access Point` message will **not** be prompted when performing mainframe-related operations in **IBM AD Build Configuration**.
3. Save and close the `zOS-Data.ini` configuration file.

Example of a `zOS-Data.ini` configuration file

```
[zOS_A]
use_cred=Y

[zOS_C]
use_cred=y

[zOS_D]
use_cred=N
```

Where:

- **zOS_A**, **zOS_B**, **zOS_C** and **zOS_D** are the names of the z/OS mainframe instances.
- The feature is available only for **zOS_A** and **zOS_C**.

Using the z/OS User Authentication feature

Once the z/OS mainframe instance is configured with TLS and declared in the `z0S-Data.ini` configuration file, in **IBM AD Build Configuration** you can perform the following operations:

1. **Retrieve source code information**
   a. In **IBM® Application Discovery Build Configuration**, select **Projects** tab, select the project, select the z/OS node, right-click to display the menu, and select the **Retrieve source code information** option.

      Note: You can run this operation on several projects simultaneously through several z/OS connections. Select all the z/OS connections that you want to use (several connections that are defined for the same project or for different projects) then right-click to display the menu and select **Retrieve source code information**.

   b. The `Please enter the z0S User/Password in order to acces the z0S Access Point` message is prompted when performing the **Retrieve source code information** operation.

   c. Enter the username and password for the related z/OS mainframe instance. For more information, see "z/OS Username and Password Guidelines" on page 238.

      Note: In case that you run this operation on several projects simultaneously through several z/OS connections, the username and password are requested for each mainframe instance.

   d. Perform all the necessary actions related to the **Retrieve source code information** operation. For more information, see the "Retrieving Source Code Information" on page 233 section.

2. **Retrieve operational information**
   a. In **IBM® Application Discovery Build Configuration**, select the **Projects** tab, select the project, select the z/OS node, right-click to display the menu, and select the **Retrieve operational information** option.

      Note: You can run this operation on several projects simultaneously through several z/OS connections. Select all the z/OS connections that you want to use (several connections that are defined for the same project or for different projects) then right-click to display the menu and select **Retrieve operational information**.

   b. The `Please enter the z0S User/Password in order to acces the z0S Access Point` message is prompted when performing the **Retrieve operational information** operation.

   c. Enter the username and password for the related z/OS mainframe instance. For more information, see "z/OS Username and Password Guidelines" on page 238.

      Note: In case that you run this operation on several projects simultaneously through several z/OS connections, the username and password are requested for each mainframe instance.

   d. Perform all the necessary actions related to the **Retrieve operational information** operation. For more information, see the "Retrieve Operational Information" on page 228 section.

Using IBM AD Build Client

Following is an overview of the use of IBM AD Build Client. It introduces the concepts and capabilities of the product and describes the typical sequence of tasks to be followed for setting up a project and undertaking the analysis.

Projects, folders, and files

Organizational entities for working with IBM AD Build Client include projects, folders, and source files. A project corresponds to an application.

A project contains a number of folders, where each folder refers to a specific type of source file that is used by the application. The default folders for a project are determined by the project definition at creation time. For example, a COBOL project has by default folders for COBOL, Copy, BMS, JCL, and Configuration source files. A Natural project has by default folders for Natural programs, Natural Include,

Natural Maps and Data Area. Each folder contains a list of the files of the corresponding type that are used by the original application. These files are also used by IBM AD Build Client.

Although for each IBM AD Build Client project folder a physical folder is created automatically under the project folder on the disk, any file can be added through IBM AD Build Client to the project folder without having to physically copy it to the corresponding folder on the disk. The physical folders are created only at the default location where IBM AD Build Client looks for files when you add files to the project folders. Files in an IBM AD Build Client project folder are references to the original source files somewhere on the disk or on a remote network drive and not physical copies of them.

Note: Starting with the 5.0.4 release, additional folders of specific type can be manually added to a project, if the Extensibility features have been enabled.

UI Reference

Following chapters contain detailed information about all aspects of IBM AD Build Client application. It describes the IBM AD Build Client main screen, the menus, and toolbar options. Furthermore, it contains a complete description of all the IBM AD Build Client operations.

Main Screen

The main screen that opens when the program is started contains the following elements:

- **Title** Bar.
- **Menu** Bar.
- **Status** Bar.
- The **Project** pane on the left between the toolbar and the status bar.
- The **Display** area on the right between the toolbar and the status bar.
- The **Output** pane across the width of the screen under the **Project** pane and **Display** area.

Main Menu

IBM AD Build Client operations are controlled by choosing commands on the main menu and menus, clicking icons on the toolbar, and keyboard shortcuts. The **Main Menu** commands are summarized in the following table. Equivalent keyboard shortcuts, when available, are also listed.

File	Keyboard shortcut	Description
New Text File	CTRL+N	Creates and opens a new text file.
New Project		Creates and opens a new project.
Open	CTRL+O	Opens the **Windows Open** dialog box, from which any file can be selected and opened.
Close		Closes the active window.
Open Project		Selects and opens an existing **IBM AD Build** project.
Save Project		Saves the current project.
Close Project		Closes the current project.
Save	CTRL+S	Saves the active window.
Save As		Saves the active window under a new name.
Save All		Saves all components of the project.
Print Setup		Opens the **Windows Print Setup** dialog box.
Recent Files		Lists the last six files opened.

File	Keyboard shortcut	Description
Recent Projects		Lists the last six projects opened.
Exit		Exits **IBM AD Build**.

Edit	Keyboard shortcut	Description
Paste	CTRL+V	Pastes the clipboard text to the cursor position.
Find	CTRL+F	Finds the string that is specified in the **Find** command.
Find next	F3	Finds the next occurrence of the string that is specified in the previous **Find** command.
Go To	Go To	Places the cursor at the beginning of the specified line number.

View	Keyboard shortcut	Description
Toolbar		Toggles on/off the toolbar.
Status Bar		Toggles on/off the status bar.
Project	ALT+0	Toggles on/off the project pane.
Output	ALT+2	Toggles on/off the output pane.
Options		Opens **Options** dialog box where you can specify the output parameters.

Project	Keyboard shortcut	Description
Add Files		Adds files to a folder in the active project.
New Folder		Create a folder in the active project and allows specifying the types of files it contains.
Settings		Opens the **Settings** window.

Build	Keyboard shortcut	Description
Make Project	F7	Similar to **Build Project**, but it creates a build operation on components of the project that are modified since the last build was ran.
Build File		Builds the current file.
Build Project	CTRL+B	Builds all files in the active projects.
Stop Build		Stops the current build.
Decisions		Opens the **Decisions** window.

Window	Keyboard shortcut	Description
Cascade		Arranges windows one behind the other in the display area.
Tile Horizontally		Displays all windows, arranged horizontally.
Tile Vertically		Displays all windows, arranged vertically.
Arrange Icons		This option is not currently available.

Help	Keyboard shortcut	Description
About IBM Application Discovery Build		Provides the current **IBM AD Build** version and information on how to access technical support.

Main Screen Toolbar

The main screen toolbar icons enable frequently used menu commands to be run without having to browse through the menu hierarchy. A brief explanation of each is presented in the following table.

Icon	Function	Menu Bar/ keyboard shortcut Equivalent	Explanation
	New File	File / New / Text File Ctrl+N	Creates a text file and opens it.
	Save File	Save File	Saves the current file.
	Save All	File / Save All	Saves the current state of the project and files.
	Previous Window	N/A	Displays the previous window
	Next Window	N/A	N/A
	Print	File / Print Ctrl+P	Prints the selected/displayed file.
	New Project	File / New / New Project	Creates a project.
	Open Project	File / Open Project	Opens an existing project.
	Check Project	N/A	Checks the active selected project for errors.
	Build Files	Build / Rebuild File	(Re)Builds the currently selected files.
	Build Project	Build / Rebuild Active Project	(Re)Builds the active project.
	Stop Build/Check	Build / Stop Build / Check	Stops the current build/check process.
	Make Project	Build / Make Project	Similar to **Build**, runs a build operation only on project parts, which are updated since the last **Build** was run.

Project Tab

The **Project** tab displays tree hierarchy of objects in the project. The tree can be expanded or collapsed by clicking the **+** or **-** signs to the left of each node. The type of each branch is identified by an icon and a text label. In most cases, a node corresponds to a specific line of code in one of the project files and double-clicking the node causes the source file to be displayed in an edit window with the corresponding code line highlighted. Right-clicking a node causes a menu to open, which usually contains commands for displaying the code (similar to the double-clicking the node), for expanding or collapsing the branch represented by the node, or viewing properties of the object.

The **Project** tab contains the following nodes under the main project node:

Node Name	Icon	Description
File Type Folder Node		Each file type folder represents a logical container for source files of the corresponding type that are included in the project. The files list is displayed when the node is expanded.
COBOL Node		Opens the COBOL source file.
Include (Copy) Node		Opens the Include (Copybook) file
BMS Node		Opens the BMS file.
JCL Node		Opens the JCL file.
Configuration Node		Opens a configuration file.

Tab Icons Summary

The following table summarizes the icons that are used in the Project pane:

Icon	Explanation
	Project
	Folder (file type)
	Program file
	Include file
	BMS screen
	JCL
	Configuration file
	Schema (closed)
	Schema (opened)
	Natural Map file
	Copy file

Icon	Explanation
	Screen file
	Printer file
	Object listing / Datasets definition Table definition / Scheduling information / Batch Processes information

Right Click / Shortcut Menus

When you right-click in different locations in **IBM AD Build Client**, different menus are available. These menus are described in the following sections.

Note: The menus might not appear exactly as described here.

Project Tab Shortcut Menu

The project tab right-click menu contains the following options:

Menu Options	Explanation
Add Files	Adds files to the folder.
Add All Files from Folder	Adds all files from the selected folder. For details on how to make this operation that is run in the background see “Adding Files to Project Folders” on page 257.
Delete All Files from this Virtual Folder	Deletes all the files from the current virtual folder.
Add Files from Mainframe Library	Adds files to the folder from the mainframe library. Mainframe libraries are available if **IBM AD Connect for Mainframe** was used previously, by using the **IBM AD Build Configuration (z/OS)**, to scan source libraries on the mainframe. For more information, see “Adding Files From Mainframe Library” on page 263 in **Tasks** and “Bringing data from mainframe libraries (PDS Libraries, Endevor, Librarian, Natural)” on page 233 in **z/OS** tab from **IBM AD Build Configuration**.
New Folder	Creates a folder in the active project. The new folder can have one file type only, which is the same or a subset of the parent folder.
Build	Builds the selected files and folders.
Delete	Deletes the selected folder (only folders, which are not the default ones that are created at project creation time can be deleted).
Settings...	Opens the **Settings** window, focusing on the folder’s settings.
Expand	Expands the folder.
Collapse	Collapses the folder.
Properties	Displays folder properties.

Project Node Shortcut Menu

The project node right-click menu contains the following options:

Menu Option	Explanation
Check	Checks for components that are referenced in the project source code, but missing from the project definition. This option is used to ensure project completeness.
Build	(Re)Builds the project.

Menu Option	Explanation
Build imposed selection	Builds the selected resources and folders.
Make	Similar to **Build Project**, but only performs a build operation on components of the project, which are modified since the last build was run.
Update API Resolution	This option allows the user to run the resolving mechanism of API calls in case that JSON resolutions, present in the User Exists Config JSON configuration file, have been modified.
Update Modified Mainframe Members/ Synchronize Members	Updates the resources that are brought in the project from the mainframe and that are changed since the last build. If **Enable Members Synchronization** option is selected, **Update Modified Mainframe Members** changes into **Synchronize Members**. For more information, see “Updating Projects” on page 266 and “Synchronizing Mainframe Members” on page 266.
New Folder	Defines a new folder, all file types are available for a folder under the project root node.
Delete	Deletes the selected file. When this option is selected, a confirmation message appears asking you to confirm or cancel the delete operation.
Settings...	Opens **Settings** window, focusing on the whole project settings.
Expand	Expands the project tree.
Collapse	Collapses the project tree.
View Repository	This option is not currently available.
Search in Tab (Ctrl+Q)	Searches within the current tab for the specified string.
Search in Tab next	Not available in the current version.
Properties	Displays project properties.

White Space Shortcut Menu

The **White Space** menu appears when you right-click anywhere in the white space of the **Project** pane.

Note: The project tree needs to be collapsed to display the menu.

The white space right-click menu contains the following options:

Menu Options	Explanation
Docking View	Docks/undocks the pane.
Hide Workspace / Hide Window	Hides the **Project** pane. Use **View** / **Project** or **Alt-0** to display it again.

Editing Shortcut Menu

The **Editing** menu contains standard editing commands (**Undo**, **Cut**, **Copy**, **Paste**) and appears when you right-click from within a text file (program).

Output Pane

IBM AD Build Client displays progress and error messages in the **Output** pane. By default, the pane is docked across the entire width of the **IBM AD Build Client** main window. It can be undocked by

double-clicking its window border, and docked again by dragging it down. The **Docking View** toggle option is also available on the menu. Double-clicking the name of a resource from the **Output** pane opens the resource in the Editor.

Output Pane Shortcut Menu

Menu Option	Explanation
Copy	Copies the selected text in the output pane to the clipboard.
Clear	Clears the **Output** pane.
Hide	Hides (closes) the **Output** pane. Use **View** / **Output** or **ALT+2** to open it again.
Docking View	Switches the window between docked and undocked states. In the docked state, the window is positioned along the entire width of the window (or it is minimized if the status bar is hidden). When the window is undocked, it behaves as a standard **Windows** window.
Save Output File	Saves the current contents of the **Output** pane to a file. The standard **Windows Save As** dialog box opens for specifying the name and location of the file to be saved.
Go to Error	When an error message in the **Output** pane is highlighted (by clicking it), this menu opens the corresponding source file in an **Editor** window at the statement that caused the error. The file can also be opened by double-clicking the error message directly.

Tasks

Working with IBM AD Build Client usually includes the following tasks:

Task	Explanation
1. Starting **IBM AD Build Client**	
2. Starting **IBM AD Build Configuration**	Define the database connection parameters (if applicable).
3. Create Project	Create a project by specifying the project name, location, environment, languages, DB type, and Map type and the relational database server name.
4. Project Settings	Adjust the project settings.
5. Project Files	Add files to the project.
6. (Re)build	Build the project.

Tasks 2, 3, 4 and 5 are set up and organizational steps. Their purpose is to define the source material to be analyzed. Step 6 (Build) creates and populates the repository, which is the basis for the Analysis step in **IBM AD Analyze**.

The following sections describe the typical tasks that are run in **IBM AD Build Client**. In many cases, you have alternative ways for activating the same **IBM AD Build Client** functions (main menu, menus, keyboard shortcuts, and the main screen toolbar). In **Tasks**, references are mostly made to the main menu commands. The alternatives are described in “UI Reference” on page 241.

Starting IBM AD Build Client

When **IBM AD Build Client** is started, the main screen appears. All activity takes place within this screen. It is empty until a new project is created or an existing one loaded.

Creating a project

Follow the steps in the corresponding topic to create a project.

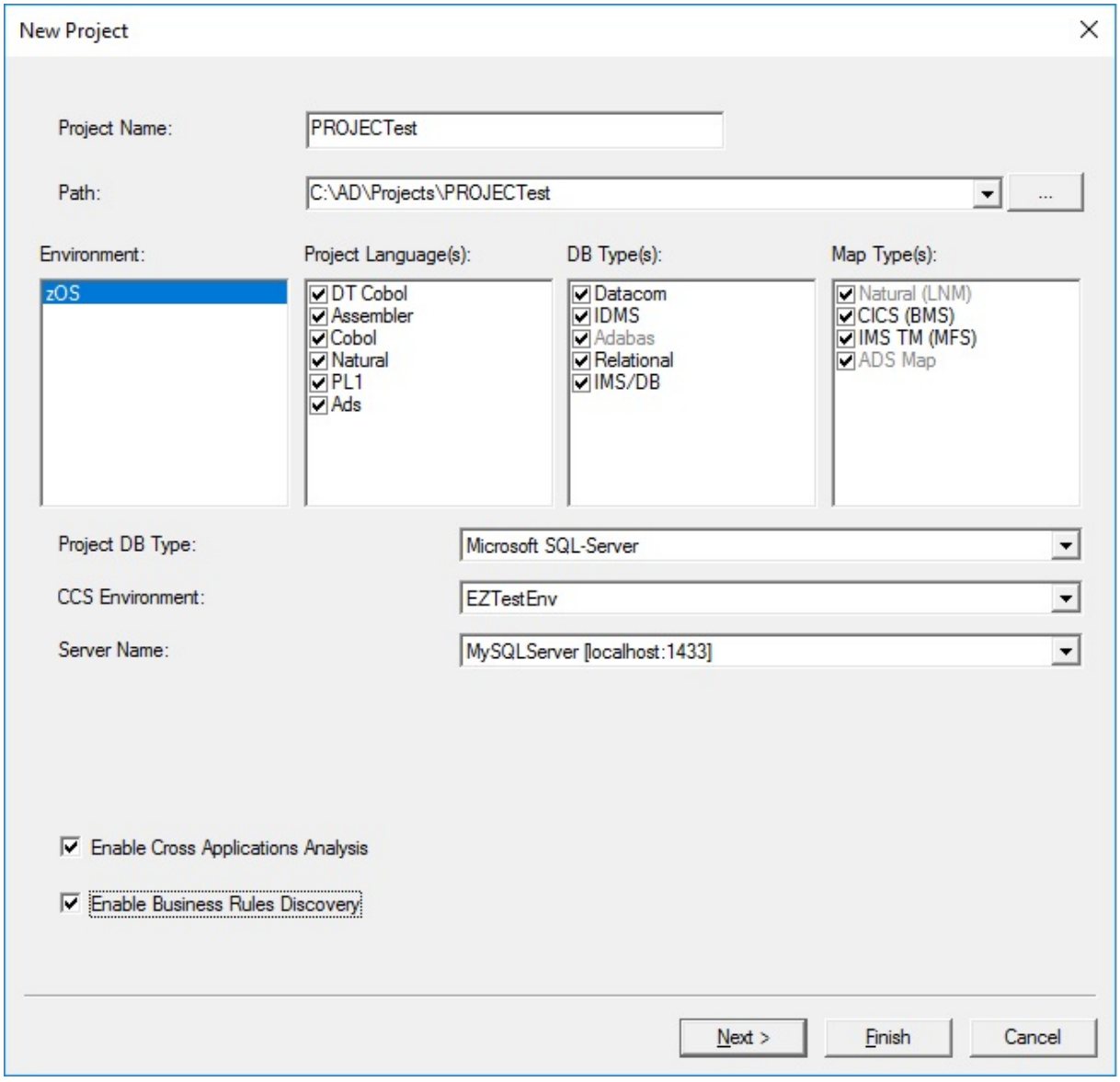
Creating a project with a Microsoft SQL server database

IBM AD Build projects correspond to independent applications. An IBM AD Build project can contain references to all application source files or to part of them. The source files are organized into folders that are category lists for the different kinds of files that make up the project/application. For example, program (such as COBOL) source, copy, and BMS files are listed in the project's Program, Copy, and BMS folders. Standard folders are defined and included in the project by default. However, you can define new folders if necessary.

Creating an IBM AD Build project creates a project folder on the computer or on a network drive. You can specify the location for this folder.

To create a new project, follow these steps.

1. Select **File** > **New** > **New Project**. The **New Project** window appears.

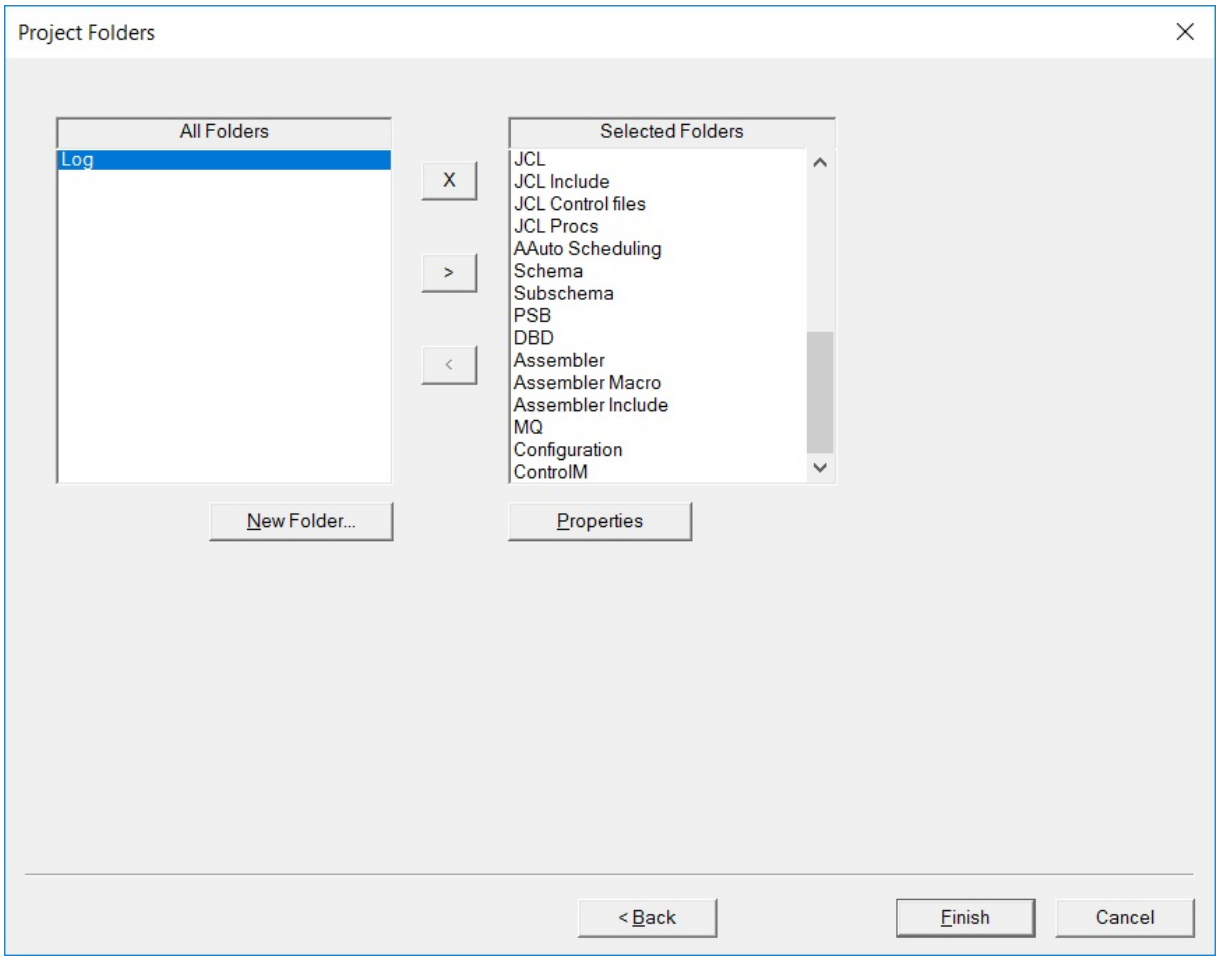


Note: The options available in the **New Project** window depend on the version of the purchased application.

2. Enter the name of the new project in the **Project Name** field.
3. The **Path** field displays the default projects path. To select a different path, click **Browse** and select an alternative location.
4. The Environment, Project Languages, DB Types, and Map Types sections present the default options.

5. From the **Project DB Type** list, select **Microsoft SQL-Server**.
6. Select an environment from the **CCS Environment** list. The environment options are defined in IBM AD Configuration Service. For more information about environments, see “Adding an Environment” on page 38.
7. Select a server from the **Server Name** list. The options are associated to the selected environment that is defined in IBM AD Configuration Service.
8. To enable the Business Rules Discovery (BRD) and the Cross Applications Analysis features at project creation time, select the related check boxes.

   After a project is created, the BRD feature can still be enabled. For more information, see “Enabling Business Rules Discovery” on page 48.
9. Click **Next**. The **Project Folders** dialog box appears for selecting and defining project folders. This screen presents different folder names, depending on the environment selected.



10. To accept the default folders without entering the **Project Folders** screen at all, click **Finish** instead of **Next**.
11. Select folders by moving them from the **All Folders** to the **Selected Folder** lists, by using **>**, or clear them using **<**. Default **All Folders** and **Selected Folder** lists are provided. The content of these lists depends on the project type that is selected in the previous step. For an existing project, you can select **Project** > **New Folder** to open a dialog box for adding more folders.
12. Click **Finish**. The new project is created and displayed as a tree in the **Project** pane (left side of the window).

Creating a project with a Db2 for z/OS database

IBM AD Build projects correspond to independent applications. An IBM AD Build project can contain references to all application source files or to part of them. The source files are organized into folders that are category lists for the different kinds of files that make up the project/application. For example,

program (such as COBOL) source, copy, and BMS files are listed in the project's Program, Copy, and BMS folders. Standard folders are defined and included in the project by default. However, you can define new folders if necessary.

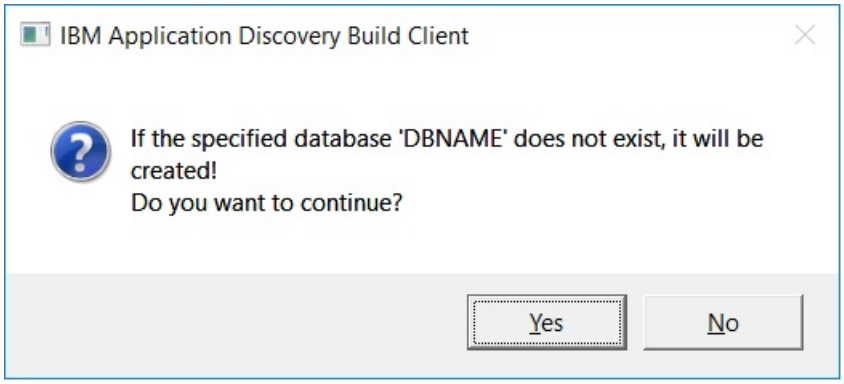
Creating an IBM AD Build project creates a project folder on the computer or on a network drive. You can specify the location for this folder.

To create a new project, follow these steps.

1. Select **File** > **New** > **New Project**. The **New Project** window appears.

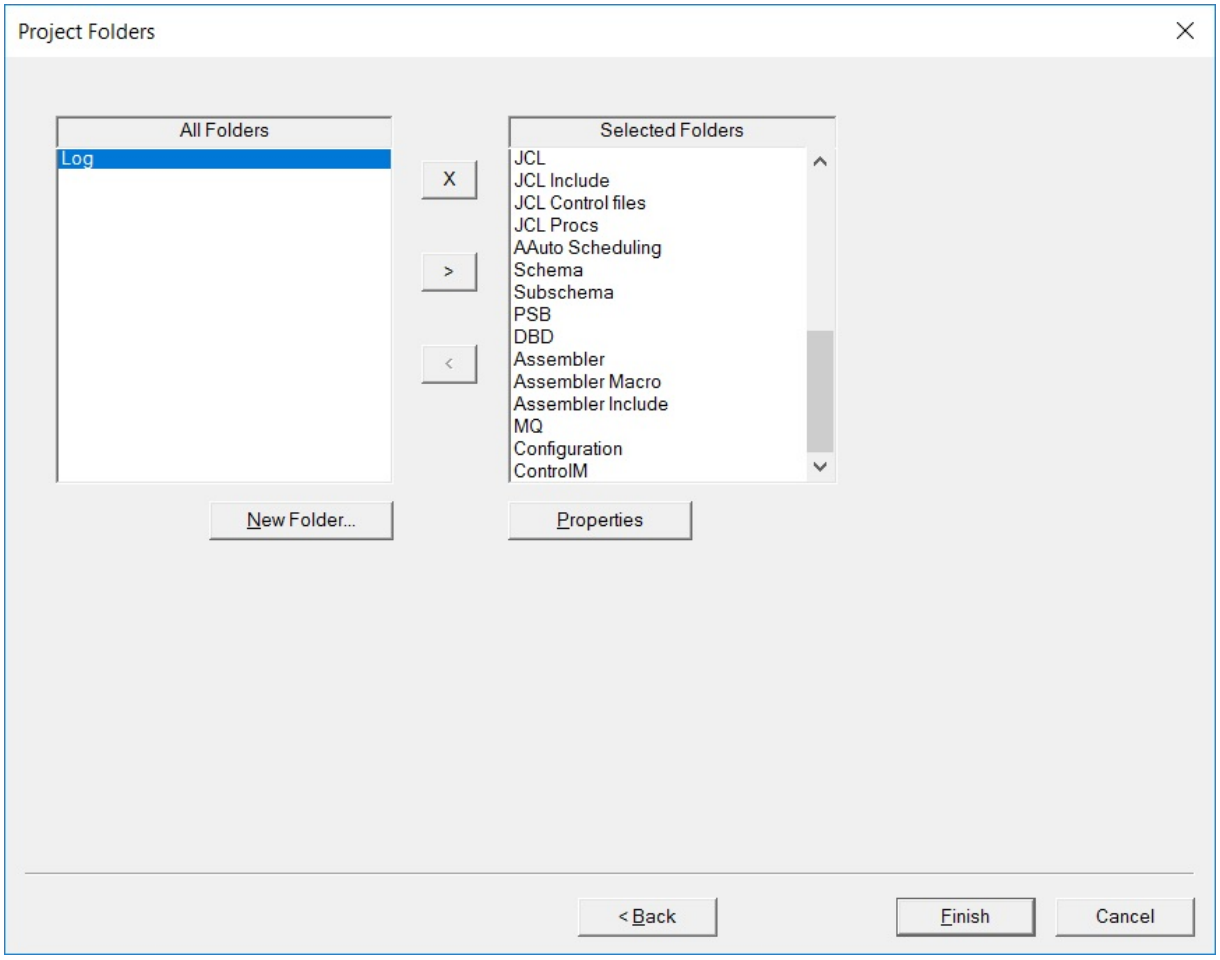
Note: The options available in the **New Project** window depend on the version of the purchased application.

2. Enter the name of the new project in the **Project Name** field.
3. The **Path** field displays the default projects path. To select a different path, click **Browse** and select an alternative location.
4. The Environment, Project Languages, DB Types, and Map Types sections present the default options.
5. From the **Project DB Type** list, select **IBM DB2/zOS**.
6. Select an environment from the **CCS Environment** list. The environment options are defined in IBM AD Configuration Service. For more information about environments, see "Adding an Environment" on page 38.
7. Select a server from the **Server Name** list. The options are associated to the selected environment that is defined in IBM AD Configuration Service.
8. **Note:**
   - IBM AD Build supports multiple projects in one database. To specify the database name and schema name select the **DB2 Details** checkbox. A new project can be created in an existing database with a different schema name. When you click **Finish** the following pop-up appears. If the database that is present in the **DB Name** field does not exist it is automatically created when you click **Yes**.



   - A new Db2 database and schema can be created by using `DB2_CreateObjects.sql` DDL script. The database can be attached by using the **Attach to database** option where the database name and schema name need to be introduced manually. The **Attach to database** option is available only when the **DB2 Details** checkbox is selected. For more information, see "Creating Db2 Database Using DDL Script" on page 285.
9. To enable the Business Rules Discovery (BRD) feature at project creation time, select the related check box.

   After a project is created, the BRD feature can still be enabled. For more information, see "Enabling Business Rules Discovery" on page 48.
10. Click **Next**. The **Project Folders** dialog box appears for selecting and defining project folders. This screen presents different folder names, depending on the environment selected.



11. To accept the default folders without entering the **Project Folders** screen at all, click **Finish** instead of **Next**.
12. Select folders by moving them from the **All Folders** to the **Selected Folder** lists, by using **>**, or clear them using **<**. Default **All Folders** and **Selected Folder** lists are provided. The content of these lists depends on the project type that is selected in the previous step. For an existing project, you can select **Project** > **New Folder** to open a dialog box for adding more folders.
13. Click **Finish**. The new project is created and displayed as a tree in the **Project** pane (left side of the window).

Creating a project with a Db2 for LUW (Linux, UNIX, and Windows) database

IBM AD Build projects correspond to independent applications. An IBM AD Build project can contain references to all application source files or to part of them. The source files are organized into folders that are category lists for the different kinds of files that make up the project/application. For example, program (such as COBOL) source, copy, and BMS files are listed in the project's Program, Copy, and BMS folders. Standard folders are defined and included in the project by default. However, you can define new folders if necessary.

Creating an IBM AD Build project creates a project folder on the computer or on a network drive. You can specify the location for this folder.

To create a new project, follow these steps.

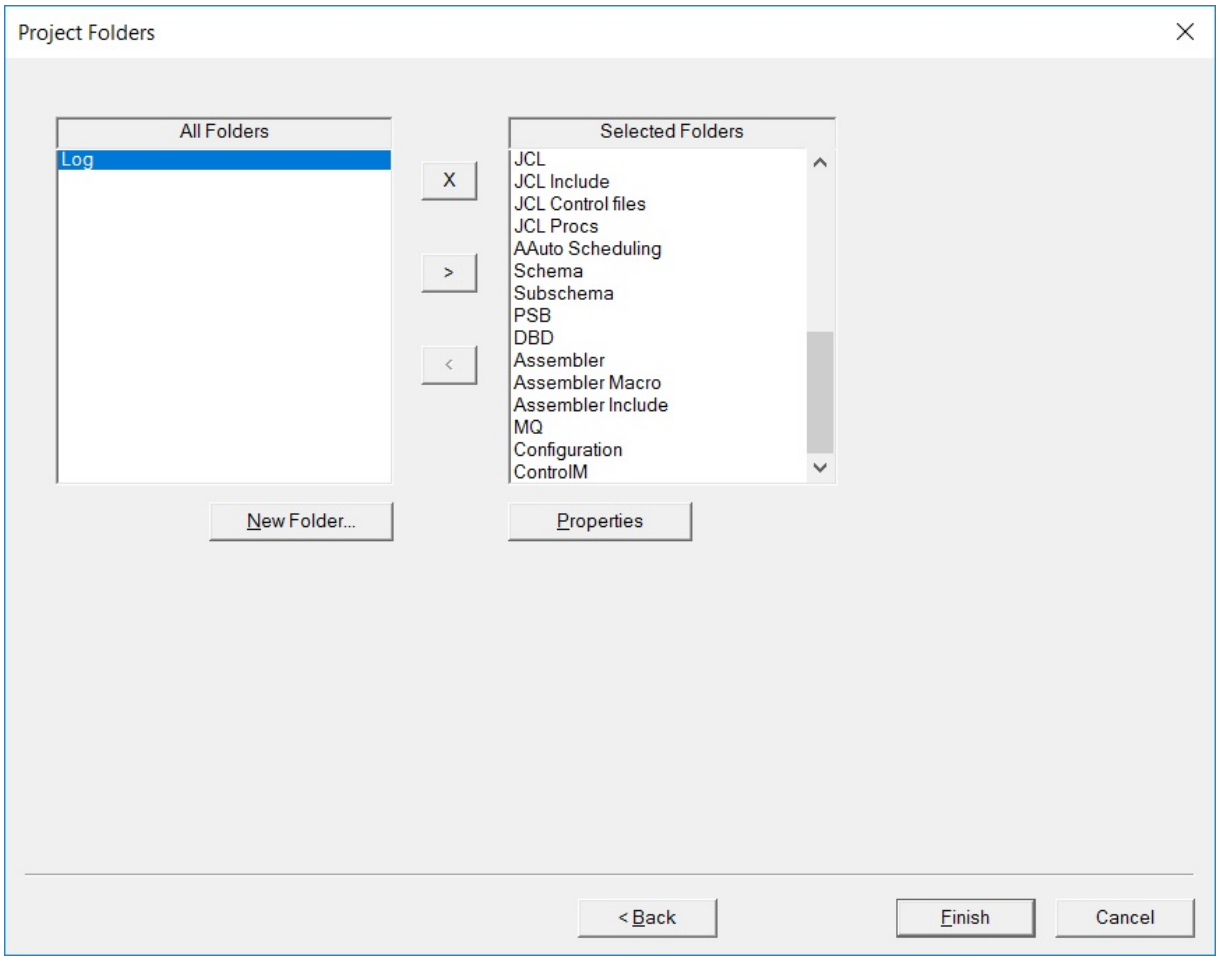
1. Select **File** > **New** > **New Project**. The **New Project** window appears.

Note: The options available in the **New Project** window depend on the version of the purchased application.

2. Enter the name of the new project in the **Project Name** field.

3. The **Path** field displays the default projects path. To select a different path, click **Browse** and select an alternative location.
4. The Environment, Project Languages, DB Types, and Map Types sections present the default options.
5. From the **Project DB Type** list, select **IBM DB2 LUW**.
6. Select an environment from the **CCS Environment** list. The environment options are defined in IBM AD Configuration Service. For more information about environments, see “Adding an Environment” on page 38.
7. Select a server from the **Server Name** list. The options are associated to the selected environment that is defined in IBM AD Configuration Service.
8. You must specify a database name in the **DB Name** field. The name must follow the general Db2 object naming rules. If this database has been created by using the `DB2LUW_CreateDatabase.sql` or `DB2LUW_CreateObjects.sql` DDL script, select the **Attach to database** check box. Otherwise, do not select the check box; when creating the project, a new database will be created with the specified name. The DDL scripts can be found in *<IBMADBuildClient_installation_path>*`\bin\release\DBScripts\`. For more information, see topic “Creating Db2 Database Using DDL Script” on page 285.
9. To enable the Business Rules Discovery (BRD) and the Cross Applications Analysis features at project creation time, select the related check boxes.

   After a project is created, the BRD feature can still be enabled. For more information, see “Enabling Business Rules Discovery” on page 48.
10. Click **Next**. The **Project Folders** dialog box appears for selecting and defining project folders. This screen presents different folder names, depending on the environment selected.



11. To accept the default folders without entering the **Project Folders** screen at all, click **Finish** instead of **Next**.

12. Select folders by moving them from the **All Folders** to the **Selected Folder** lists, by using **>**, or clear them using **<**. Default **All Folders** and **Selected Folder** lists are provided. The content of these lists depends on the project type that is selected in the previous step. For an existing project, you can select **Project** > **New Folder** to open a dialog box for adding more folders.
13. Click **Finish**. The new project is created and displayed as a tree in the **Project** pane (left side of the window).

Creating a Project by using the Export Path CSV File

Starting with **IBM AD V5.1.0.7**, the new Java utility `com.ibm.ad.create.project-1.0.0.jar`, present in the `<IBM ADDI Installation Folder>/support/CreateADProject` folder helps you to create a new project in the background.

1. Before you run the `com.ibm.ad.create.project-1.0.0.jar` Java utility, perform the following steps:

   a. Copy the exported `.CSV` file under `/CreateADProject` folder.

   Note: The `.CSV` file represents the output of the **Export Paths** feature, file that is generated from **IBM AD Analyze Client**. For more information, see "Graphical Analysis Overview" on page 321.

   b. Define the content of the `NPSkel.ini` file as follows:

```
ProjectName = "%%ProjectName%%"
Path = "<REPLACE_WITH_PATH_TO_AD_PROJECTS_FOLDER>\%%ProjectName%%"
Environment = "zOS"
ProjectLanguages = "DT Cobol,Assembler,Cobol,Natural,PL1,Ads"
DBTypes = "Datacom,IDMS,Adabas,Relational,IMS/DB"
MapTypes = "Natural (LNM),CICS (BMS), IMS TM (MFS),ADS Map"
ProjectDBType = "SQL" <OR> "DB2"
CCSEnvironment = "<REPLACE_WITH_ENVIRONMENT_NAME_FROM_CCS>"
DBServerName = "<REPLACE_WITH_DB_SERVER_NAME_FROM_CCS>
[localhost:<REPLACE_WITH_DB_PORT_FROM_CCS>]"
```

   Note: For more information on how to define the `NPSkel.ini` file, see "1. Creating a new project in background" on page 297.

2. Run the following command:

```
java -jar com.ibm.ad.create.project-1.0.0.jar "C:\Program Files\IBM Application
Discovery and Delivery Intelligence\IBM Application Discovery Build
Client\Bin\Release\IBMApplicationDiscoveryBuildClient.exe" ProjectName ExportedFile.csv
```

   Where:

   - **ProjectName** - represents the name of the project that will be created.
   - **ExportedFile.csv** - represents the output of the **Export Paths** feature.

3. As a result, a new project is created. It contains all information that is defined in the `NPSkel.ini` and `.CSV` files.

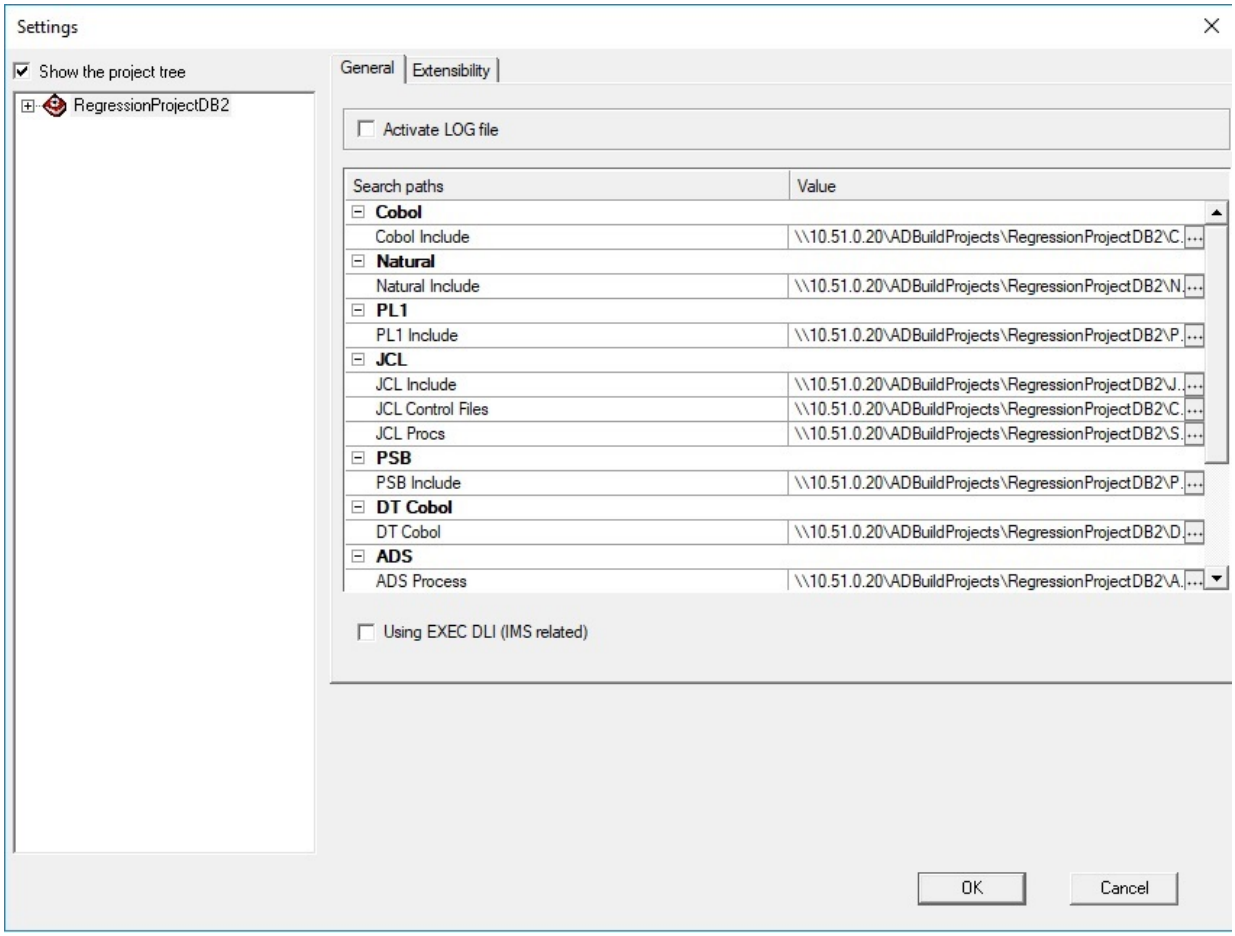
   Note: The created project is already built.

Adjusting Settings

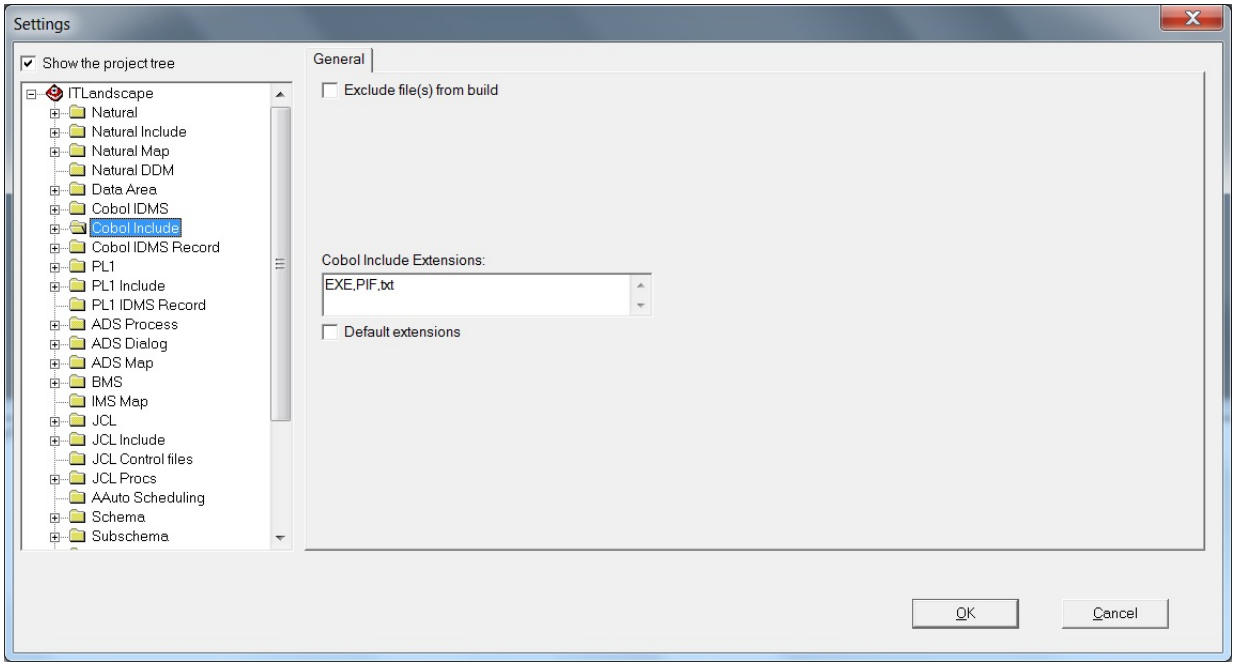
From the **Settings** window, the following actions can be performed.

1. From the **Search Paths Order** window, modify the default search paths, add several search paths for a resource type, and set the order in which these paths are accessed.

   When resources such as COBOL, Natural, PL/I are built in **IBM AD Build Client**, the corresponding `include/copybook`, `control`, `proc`, and `macro` files are searched, according to the default extensions in the default project folders.



2. Generate a log file under each project folder during the build process. This procedure takes up more disk space but allows a detailed inspection of the build process if an error occurs. It is recommend to keep this option cleared. You may use it if detailed debug information is required.
3. Determine whether a file or all the files from a project folder is included or not in the analysis (Build). For the Include folders (`Natural Include`, `Cobol Include`, `Assembler Include`), use **Settings** to override the default extensions for these files. More parameters are available for each resource type.



4. Set up an **IMS DB Environment** for COBOL programs that use **EXEC DLI** commands and **DL/I** calls.

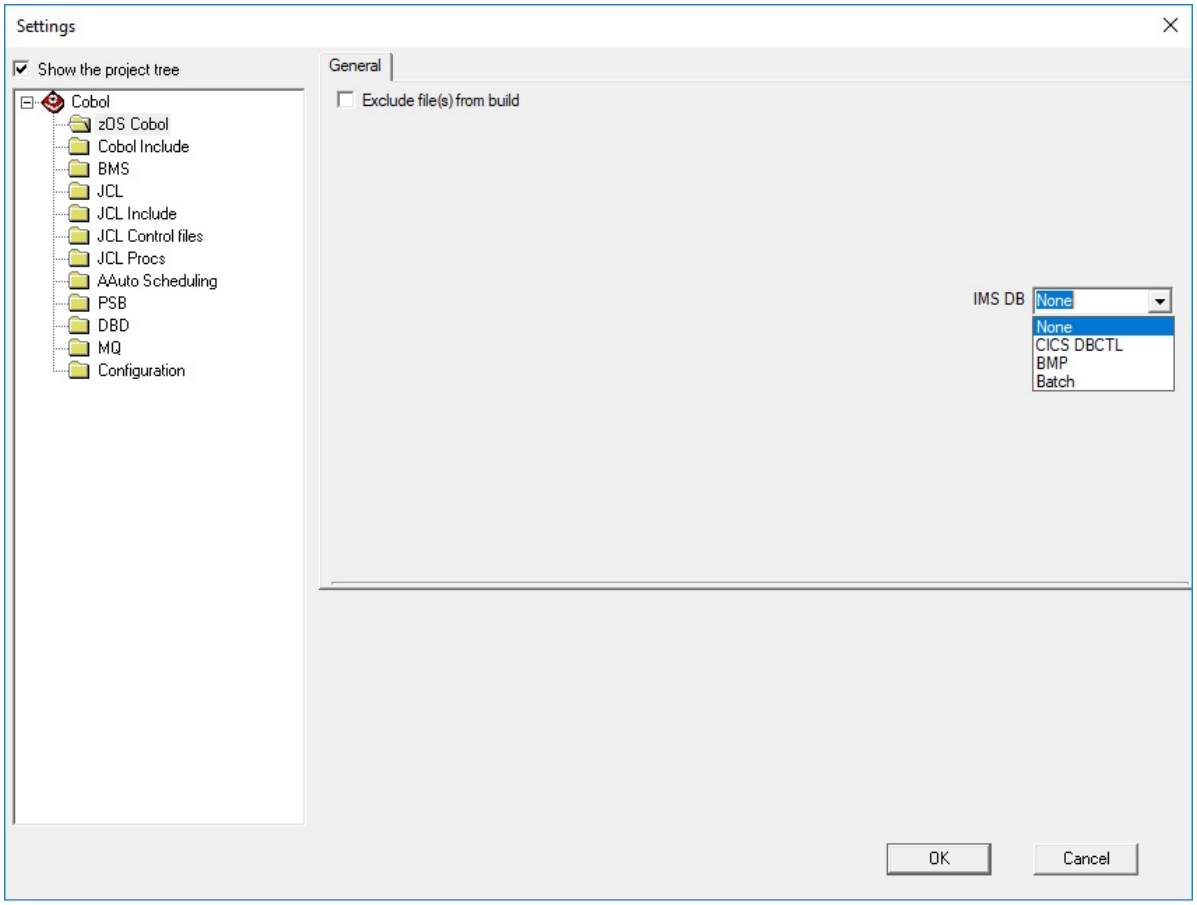
   A corresponding **IMS DB Environment** needs to be set up for the programs that access IMS databases and/or IMS transactions.

   IBM AD Build Client analyzes COBOL programs that use **EXEC DLI** commands and **DL/I** calls. All programs that access IMS databases and/or IMS transactions need to have a corresponding **PSB**, therefore an appropriate environment needs to be set up at the folder's project level.

   To set up an **IMS DB Environment**, follow these steps:

   a. Select **Show the project tree** check box and expand the project tree.

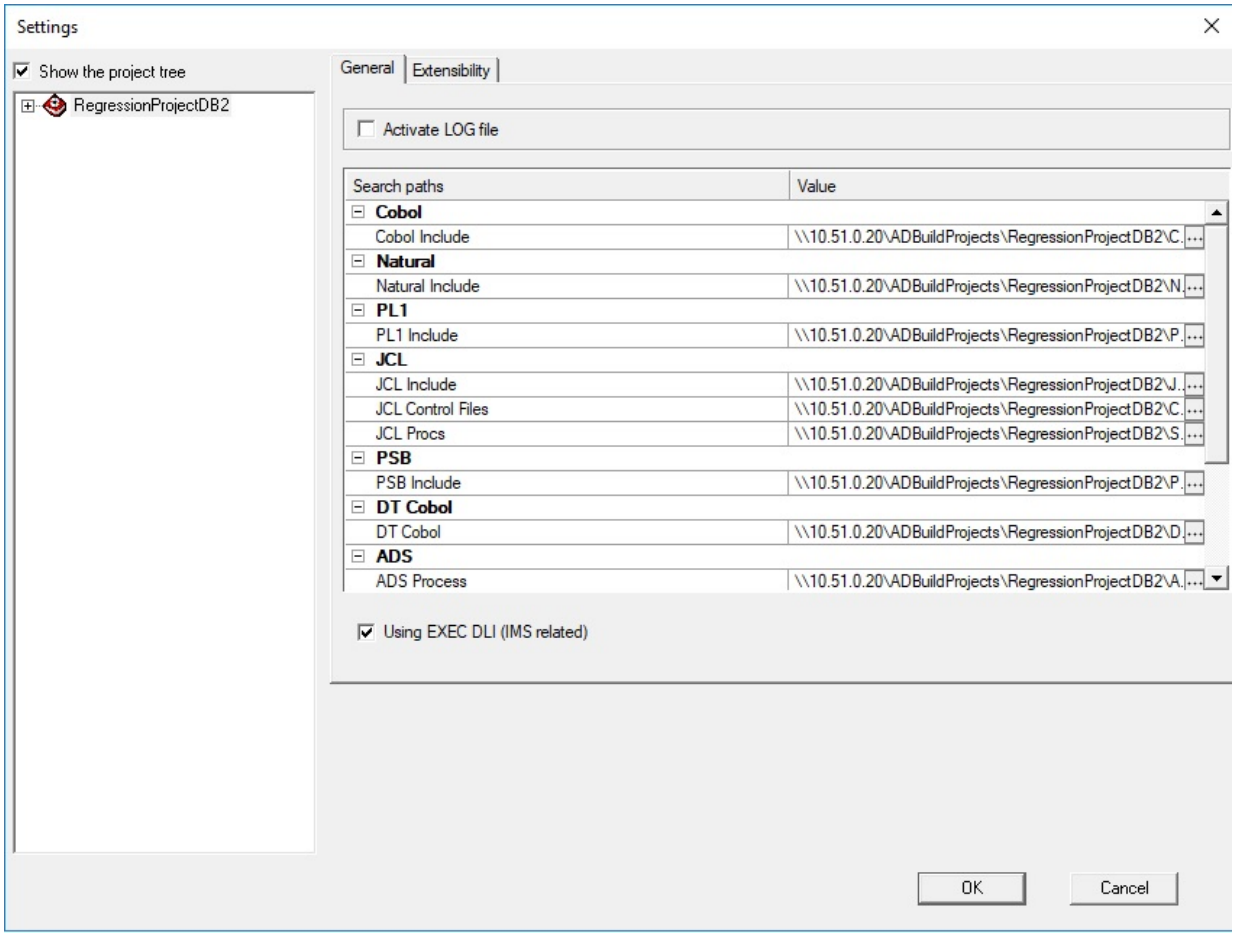
   b. Select **zOS Cobol** folder and choose the appropriate **IMS DB Environment** as in the following image.



   Note: The **None** option is selected by default.

   For more information about the difference between the environments, go to the PCBs and PSB topic in the IBM IMS documentation.

5. Select the **Using EXEC DLI (IMS related)** check box to analyze COBOL programs with **EXEC DLI** commands that are present in the project.



When **Using EXEC DLI (IMS related)** check box is selected, two builds are triggered, increasing the build operation time.

The second build is triggered when program *"A"* calls subprogram *"B"*, where *"A"* is the main COBOL program that has a corresponding PSB, and *"B"* is the subprogram that contains the **EXEC DLI** commands.

Note: As a result, a message is shown in the output window, informing that **Building programs related to IMS EXEC DLI in subprograms**.

During the first build, a `IMSExecDliInSubprograms.txt` file is generated automatically and has the following format:

```
<called program name "B">, <parameter number 1>, <OffsetStart1>, <OffsetLength1>,
<LinkageSectionVariable1>, <PCBNumber>
```

The generated file is used to resolve the parameters that are parsed from the main COBOL program *"A"* to the subprogram *"B"*.

Examples of the generated `IMSExecDliInSubprograms.txt` file:

- When a program name is called together with the parameters and their positions, where -1 represents the PCB number

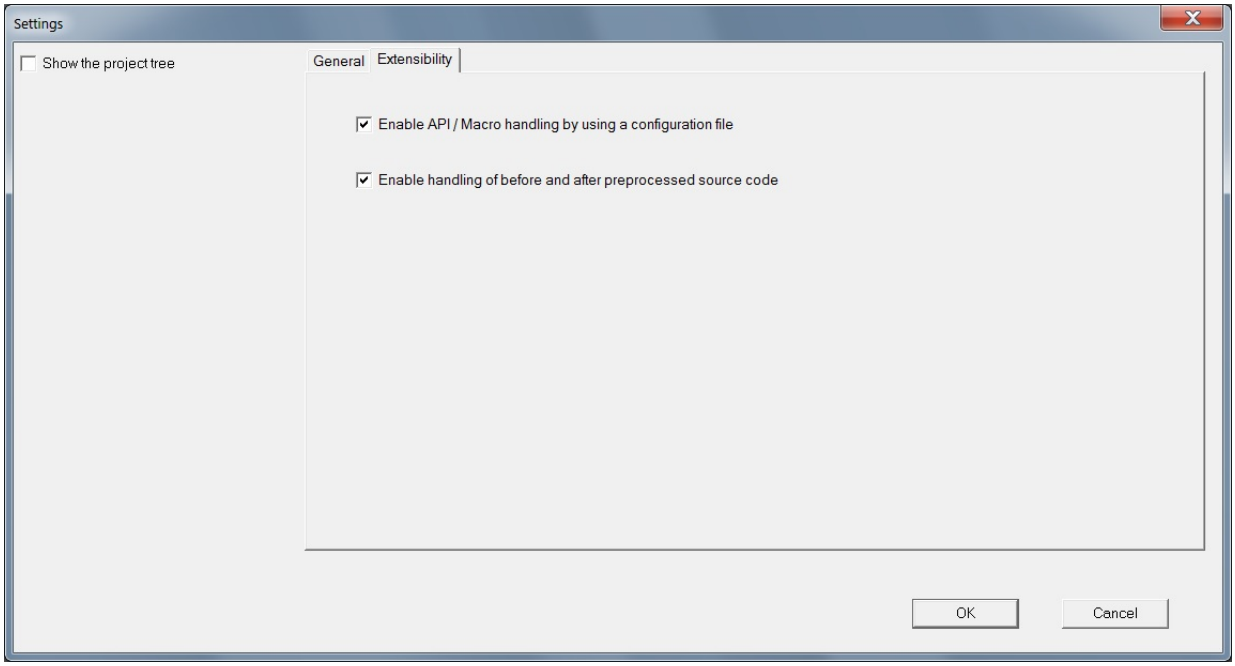
  ```
  B,4,1,2,VAR1-PCB-NUM,-1
  ```

- When a program name is called together with the PCB number, where -1 represents the parameters and their positions

  ```
  B,-1,-1,-1,,11
  ```

6. Enable the **Extensibility** features:

- **Enable API/Macro handling by using a configuration file.**
- **Enable handling of before and after preprocessed source code**.



For more information, see "Using the Settings Option" on page 273.

Adding Files to Project Folders

After you create the project and its folders, the files to be analyzed must be added to the appropriate project folders. Following are the project folders and the sources that can be placed in each one of them.

Note:

- Project folders depend on the type of selected project at project creation time. Therefore, for some projects, some of the following folders are not available.
- When adding files, if the limit of **40000** files is reached, additional virtual folders are created at the project's level. They do not have correspondent on disk and they cannot be deleted manually from **IBM AD Build Client**. The additional virtual folders are different than the existing virtual folders by having a special naming convention, suffixed by **@&@<incremental_number>**. Examples of additional virtual folders for `zOS Cobol`:

```
zOS Cobol@&@1
```

```
zOS Cobol@&@2
```

The project folders that are created, differ according to the environment selected as shown in the following table:

Environment	Folders- Description
z/OS	• `AAuto Scheduling` - A-AUTO Dataset Flag Report. • `AAuto Scheduling` - A-AUTO Scheduling programs. • `ADS Dialog` - N/A. • `ADS Map` - N/A. • `ADS Process` - ADS Process files. • `API Config` - Files containing configurations for the API calls.

Environment	Folders- Description
	• Assembler - Assembler files. • Assembler Include - "Assembler include" files. • Assembler Macro - Assembler macros. • BMS - BMS assembler definitions (relevant only for CICS® projects). • Cobol IDMS - Cobol IDMS files. • Cobol IDMS Record - Cobol IDMS Record files. • Cobol Include - COBOL copybooks and include files. • Configuration - CSD files. • Configuration - IMS/T PGM • Configuration - PGM Aliases. • Control-M - XML files (exported from Control-M Enterprise Manager) containing jobs and conditions. • Data area - Natural data area that includes Local Data Area, Parameter Data Area, and Global Data Area. • DBD - IMS DBD files. • DT Cobol - Data Type Cobol files. • DT Cobol pre-compiled - Pre-compiled data type Cobol files. • IMS MAP - MFS files (relevant for IMS projects only). • JCL - JCL Jobstream files. • JCL Control files - JCL Control files. • JCL Include - JCL Include files. • JCL Procs - JCL procedure files. • MQ - MQ configuration files. • Natural - Natural programs • Natural DDM - Natural DDM files. • Natural Include - Natural include files. • Natural Map - Natural map definitions. • PL1 - PL/I programs. • PL1 IDMS Record - PL/I IDMS record files. • PL1 Include - PL/I copybooks and include files. • PreProc Before - User's original sources. • PreProc Config - Files containing mappings between the folders of the before, meta and after files. • PreProc MetaData - Files that map the before files with after files. • PSB - IMS PSB files. • Schema - IDMS schema • Subschema - IDMS Sub-schema. • zOS Cobol - Simple Cobol files.

Some environment files are not added directly to one of the project folders. Instead, they are put under the project directory on the hard disk. The project directory is the location where the project was created, specified in the path field at project creation time. Following is a list of these environment files.

Files under the project directory on the hard disk

Control files

These files can be placed by default into the CTRL directory that is automatically created under the project directory. Control files must not have any extension in order for **IBM AD Build Client** to locate them. In case you have several control files with the same name that are taken from different libraries and used by the JCL files according to the search order, create a directory under the project directory for each library. For example, if two control files with the same name that are taken from two libraries LIB1.MYCTRL and LIB2.MYCTRL, create two directories that are named LIB1.MYCTRL and LIB2.MYCTRL under the default CTRL directory and place each procedure in the corresponding directory. The **IBM AD Build Client** JCL compiler searches for the right folder according to the search order specified in the JCL.

Note: This procedure is needed only if you have two control files with the same name, in which case they cannot be both put in the default directory CTRL.

The Control files (or the PARM files) are the source members referenced in DD cards in the format of DSN=MY.PDS.NAME(CTRLMMBR). These Control files may contain SORT parameters, or SYSIN data, or Db2 command (if in SYSTSIN card for Db2 invocation programs), all depending on the step they are used in and the DD card name.

The JCL include files are files that are included in the JCL source using the INCLUDE command, e.g.//LABEL001 INCLUDE MEMBER=INCFILE1INCFILE1 is the JCL include member. Usually these will have list of DD cards commonly used together in many JCL sources, and put into one shared file to simplify maintenance in case you want to add/remove/change a DD card. They can also contain full steps.

Procedure files (also known as PROCS)

These files, which are referenced from JCL files, can be placed by default into the SYS1.PROCLIB directory that is automatically created under the project directory. Procedure files must not have any extension in order for **IBM AD Build Client** to locate them. In case you have several procedure files with the same name that come from different libraries and used by the JCL files according to the search order, create a directory under the project directory for each library. For example, if two procedure files with the same name that are taken from two libraries LIB1.MYPROC and LIB2.MYPROC, create two directories that are named LIB1.MYPROC and LIB2.MYPROC under the project directory and place each procedure file in the corresponding directory. The **IBM AD Build Client** JCL compiler searches for the right folder according to the search order specified in the JCL.

Note: This procedure is needed only if you have two procedure files or include files with the same name, in which case they cannot be both put in the default directory SYS1.PROCLIB.

PSB files

These files, used only by the **IMS** application, must be placed in a directory that is named PSB under the project directory. This directory is not created automatically and therefore must be created if needed.

The AAuto scheduling folder

This folder can host two types of files: AAuto scheduling files and AAuto Dataset Flag report files.

Before you run the build process, make sure to set the correct type for the AAuto Dataset Flag report file: In the Project pane, right-click on the Dataset flag report file that is loaded in the AAuto Scheduling folder of the project and select **Properties**. In the **File Properties** window, verify that the **Type** is set to **AAuto Dataset Flag Report**. (the file type verification can be done either when the file is loaded in the project or at a later moment, but before the build step is run).

The CICS CSD configuration file

A CICS administrator can use the **LIST** command of CICS utility DFHCSDUP to extract CSD information into a report. The report can be stored under the Configuration virtual folder and can be added to a build project as a CSD type of file. The build process parses the CSD configuration file and stores the information into the MFCICS tables.

The name of the CSD configuration file must have maximum eight characters, because the file name is used as the CICS region name. For more information, see chapter "CICS CSD Information Handling" on page 275.

The IMS transaction mapping file

This configuration file is used to map between IMS transactions and programs. The file must be placed under the `Configuration` virtual folder in the project. The type of the file must be IMS/T PGM. See the following example of mapping configurations in an IMS transaction mapping file:

```
TRANSACTION(TRAN1) PROGRAM(PROG1) IMS-TM
TRANSACTION(TRAN2) PROGRAM(PROG2) IMS-TM
```

The Pgm_Aliases file

This configuration file for aliases is used to specify external alias names coming from outside the source files. The file will be added in the `Configuration` virtual folder, in the project, with type *PGM Aliases*. The configuration file for aliases is a comma separated file, having the following format:

```
* - a commented line starts with '*'
<optional disambiguation file path>, <procedure/program name defined in file>, <alias name 2>,
<alias name 3>
```

In case the alias name is not configured with a file path, the file format is as following:

```
<program/procedure defined in file>, <alias name 1>, <alias name 2>
* procedure name in case of PLI/I file
```

PGM Aliases (Configuration) Files Example:

```
* this is a commented line
\\shared-resoures-dir\Projects\Pgm_Alias_002\PL1\PLI1, PLI1, PLI01, PLI001
\\shared-resoures-dir\Projects\Pgm_Alias_002\PL1\PLI1_1, PLI01, PLI_1, PLI_01, PLI1
PLI2, PLI002, PLI0002
PLI3, PLI03
```

Note:

- These program aliases, **<program/procedure defined in file>**, **<alias name 2>**, **<alias name 3>**, can be declared in any order, provided **<program/procedure defined in file>** exists among the aliases names. Example: if only **<alias name 2>**, **<alias name 3>** are present and **<alias name 3>** is called, while **<alias name 2>** is not found as a program/procedure definition in any source file, then **<alias name 3>** will not be replaced with **<alias 2>** in the call.
- The first item **<optional disambiguation file path>** which is the fully-qualified-name of the file, is optional and only needed when the same alias name refers to actually different programs: in the example above, the same program alias name **PLI01** refers to two different programs, defined in two different files. If only one fully-qualified-file-names of the two will be present or the two lines meant to be told apart have no alias name in common, the fully qualified file name would not make any difference.
- If disambiguation between two alias groups is needed, the fully-qualified-file-name of a PL/I file must be added in the 1st position, as it shows in **Project** > **Properties**. Example: if the file was added with a network path, the same syntax must be used into the `PGM Aliases` file.
- After adding new alias name(s) into the `PGM Aliases` configuration file, it is recommended to (re)compile both the configuration and the PL/I files, where **<procedure/program name defined in file>** exists, in this order. Example: **configuration file** > **PL/I file**.

  Note: When (re)building the entire project, the configuration file is build first by default.

Important: Currently, the **external alias names** feature is only available for PL/I programs.

The CheckReportConfiguration file

In Check Report, programs from a z/OS system and subsystems (MQ, CICS, Db2, IMS...) are reported to be missing and it's not possible to load them in IBM AD. Actually, these programs should not be reported as missing since they are part of the system or subsystems. The missing

components that need to be excluded from the Missing Component Report can be added in a configuration file.

The file must be placed under the `Configuration` virtual folder in the project. The type of the file must be **Check Exclude**. The configuration file for Check Report is a comma separated file. It contains the name of the program, the entity type and program type. The configuration file has the following structure:

```
IDENTIFIER1, TypeOfEntity1, TypeOfProgram1
IDENTIFIER2, TypeOfEntity2, TypeOfProgram2
```

Where **TypeOfEntity** can have the following values:

```
"PGM" - PROGRAM
"INC" - INCLUDE
"TRN" - TRANSACTION
"CPY" - COPYBOOK
"CTR" - CONTROL
"JOB" - JOB
"MAP" - MAP
"PRC" - PROCEDURE
"DLG" - DIALOG
"PRS" - PROCESS
"DA"  - Data Area
```

Where **TypeOfProgram** can have the following values:

```
"COB" - COBOL
"ASM" - ASSEMBLER
"PLI" - PL1
"NAT" - NATURAL
"JCL" - JCL
"BMS" - BMS
"ADS" - ADS
"ANY" - All the types mentioned above
```

Example:

```
ICEGENER, PGM, JCL
IDCAMS, PGM, JCL
IEFBR14, PGM, JCL
TEST1, PGM, COB
JCLINC, INC, JCL
AC01, TRN, COB
AC02, TRN, ANY
COPY1, CPY, COB
INCLUDE, CPY, ANY
CTR1, CTR, JCL
MAPSET:MAP, MAP, BMS
:MAP1, MAP, BMS
PROCJCL, PRC, JCL
NCAMOD, DLG, ADS
ADSAGES, PRS, ADS
DACI, MAP, ADS
DA1, DA, NAT
LAPL1, PGM, PLI
SORT, PGM, JCL
```

The PgmModuleMap file

This file is used to map between load module and the first program that is called in the module (relevant only for batch applications). By default, **IBM AD Build Client** assumes the module name and the name of first program that is called are identical. In case they are not identical, a mapping must be described in the `PgmModuleMap.txt` file, which must be placed under the project directory. Following is an example of the file content:

```
OKC82     OKC8201
OKC75     OKC7501
OJC07     OJC0701
```

On the left side, the module name is specified and on the right side, the first program name is specified.

The PSBmap file

This file is used to link the program and the PSB file names (the format contains: PgmName, PgmType, PSBFileName).

If IMS related calls (CBLTDLI and CEETDLI for Cobol programs and PLITDLI for PL1 programs) are used, **IBM AD Build Client** assumes the program name and the name of the PSB file are identical. In case they are not identical, a PSBmap.txt file needs to be created and configured to describe the mapping between the program name and the name of the PSB file.

If EXEC DLI (IMS related) is used, **IBM AD Build Client** assumes the program includes the schedule command EXEC DLI SCHD PSB. In case that the EXEC DLI SCHD PSB command is not present in the program, a PSBmap.txt file needs to be created and configured to describe the mapping between the program name and the name of the PSB file.

Important: The PSBmap.txt file needs to be placed in the root of the project's directory, to <Project Path>\<ProjectName>\ folder. The ProjectName folder was created when the project was initially defined in **IBM AD Build Client**. It is located, by default, directly under the **Default project path** filled in **IBM AD Configuration Server** > **Install Configurations** > **IBM AD Discovery Build Client**.

Following is an example of the file content:

```
EDADL3M,Cobol,EDADL3P
EDADM2M,Cobol,EDADM2P
EDADM4M,PL1,EDADM4P
EDADN2M,PL1,EDADN2P
```

On the left side, the program name is specified, in the middle the program type (Cobol or PL1) is specified, and on the right side, the PSB file name is specified. For Cobol programs, in case the PROGRAM-ID and the file name are not identical, PROGRAM-ID name is used to map (link) the Cobol program with the PSB file name.

To add files to a folder, follow these steps:

1. In the **Project** tab, click the folder name, and then select **Project** > **Add Files**. Alternatively, right-click the folder name and choose **Add Files**. A file selection window opens.
2. Locate the files (they can be on any drive and directory) and select them individually or in groups (by using the Windows **SHIFT** key or **CTRL** key mechanism).
3. Click **OK** to add the selected files to the project. The names of the files appear in the expanded file structure in the project tree.
4. Repeat the Add Files procedure to add all necessary files to each of the project folders.
5. If you need to add a long list of files, you can use the option **Add All Files from Folder**. Selecting this option presents you with the following window:



Note: Make sure that the folder path is correct; click **OK** to add all the files from that folder to the corresponding project folder.

6. To save the programs, files, and projects in their current states, select **File** / **Save All**.

Note: It is possible that the process of adding files can take a long time during which you cannot use the application. If you need to use the application, you can run the **Add files** process in the background. To make the **Add files** operation to run in the background, follow these steps:

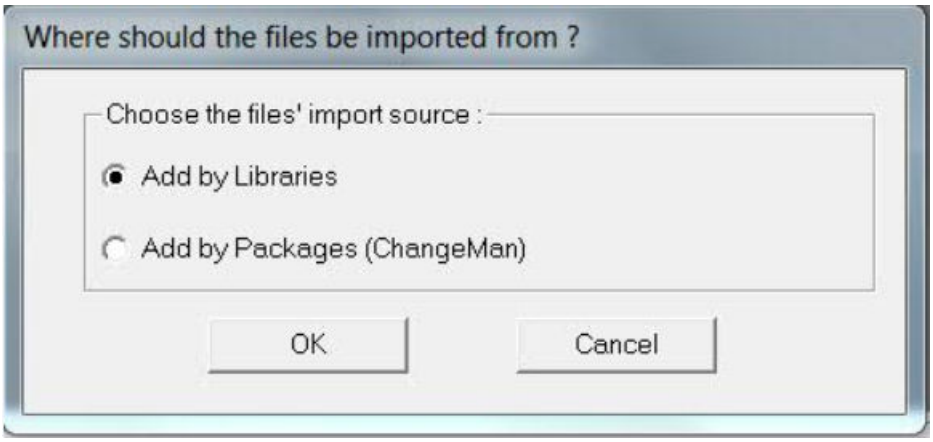
a. Click **Start**, select **Run** then type **cmd** to open the command window.

b. Go to the folder where your **IBM AD Build Client** is installed and locate the `IBMApplicationDiscoveryBuildClient.exe` file. Drag the `IBMApplicationDiscoveryBuildClient.exe` file into the command window then enter "**/?**" and press **ENTER**. A window is displayed containing detailed instructions about how to make a specific process to run in the background.

Adding Files From Mainframe Library

To add files from the mainframe library to your project, some preliminary steps need to be taken in the **IBM AD Build Configuration**. See "The zOS Tab" on page 236 for more details.

1. In your **Project** tab select the folder where you want to import files from the mainframe library then right-click to display the menu and from it select **Add Files from Mainframe** to display the following window.



2. A list of imported libraries is displayed. Select the libraries from which you want to import resources then click **Next**: the **Member Files from Mainframe Selected Libraries** window is displayed.

   Note: Only libraries that contain at least one member are displayed.

3. A list of members that are identified within the imported libraries is displayed. For each resource the following data is displayed:

   - The type of the resource (Assembler Macro CICS map BMS, Cobol Program).
   - The source (z/OS).
   - The name of the library where it was found.

4. Select the files that you want to add to your project and click **Finish**. The selected files are added in the current folder of your project: Their respective names indicate their source – z/OS, and the name of the library from where they are imported and their original name.

Adding Files From ChangeMan ZMF Packages

To add files from the mainframe by using **ChangeMan ZMF Packages**, some preliminary steps need to be taken in the **IBM AD Build Configuration**. For more information, see "The zOS Tab" on page 236 and "Configuring the z/OS Connection" on page 216.

1. In your **Project** tab, select the folder where you want to import files from the mainframe then right-click to display the menu and from it select **Add Files from Mainframe** to display the following window.



2. Select **Add by Packages (ChangeMan)** then click **OK** to display the **Add Files from Mainframe Libraries** window. A list of imported packages is displayed.

   Note: Only libraries that contain at least one member are displayed.
3. Select the package from which you want to import resources then click **Next**: the **Member Files from Mainframe Selected Libraries** is displayed. A list of members that are identified within the imported package is displayed. For each resource, the following data is displayed.
   - The method that is used for import (SRC - ChangeMan).
   - The source (z/OS).
   - The name of the package where it was found.
4. Select the files that you want to add to your project and click **Finish**. The selected files are added in the current folder of your project: Their respective names indicate their source – z/OS, and the name of the package from where they are imported and their original name.

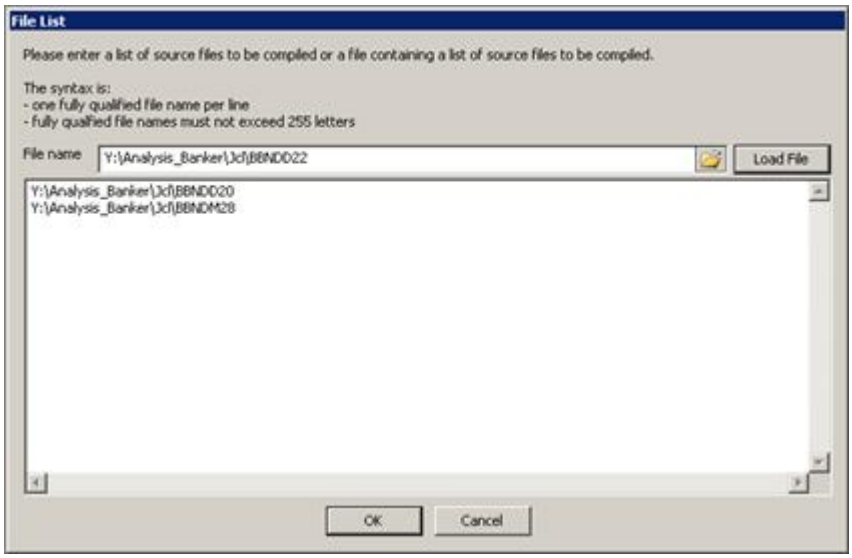
Building Projects

A “build” is the process where IBM AD Build Client reads project sources, places the results in the project repository, and generates the data that is needed to display the graphical representation of the applications’ internal and external program relationships.

The build process can be ran on individual programs in the project, on a batch of selected files and folders or globally on all the resources in the project. Generally, you make a global build, but if, for example, a single source file is changed, a build on that file alone would be appropriate. In that case, only the modified program is analyzed and the project repository is updated accordingly.

1. To build a project, follow the steps bellow.
   a) **Select Build / Build Project** to start the build process. A warning message alerts you to the fact that this operation erases the database. Click **Yes** to start the build process.
   b) As each file is processed, its name and accompanying notes and messages, including error notifications, are displayed in the **Message** pane
   c) On completion of the build, you can double-click any of these messages to open the corresponding source file at the appropriate line.
2. To build a single program or a folder, follow these steps:
   a) In the **Project** pane, expand the project tree so that the required source program or folder is visible. Click the program icon or the folder to select it then right-click and select **Build**.
   b) The **IBM AD Build Client** Message window displays the file name and log messages that are created during the build process. Information about the file and its internal relationships is created and placed into the repository.
3. To build a batch of selected files, follow these steps:
   a) In the **Project** pane, expand the project tree so that the required source programs and folders are visible. Click the programs and folders that you want to include in the build process then right-click and from the menu, select **Build**.

b) Alternatively, for large batches of files you can create a `*.txt` file that contains the list of resource files that you want to build and then use the **Build Imposed Selection** option from the project node menu to load that file.



c) Browse to the location of the `*.txt` file then click **Load** to load its contents. The `*.txt` file must contain the FULL PATH to each resource file on a separate line. Extra syntax indications for the `*.txt` file are also available. After the file is loaded, the resource files list is displayed. Click **OK** to start the build process. The **Messages** window displays the file names and log messages that are created during the build process. Information about the files and their relationships is created and placed into the repository.

4. To update the project after several sources are changed:
   a) When several sources are changed, the easiest way to update the project repository is to use the **Make** option. Run **Make** by selecting **Build / Make Project** or by pressing **F7**. **Make** works in the following way: for each source, **IBM AD Build Client** compares the last modified date with the date on the disk and decides whether an update is necessary for the source. This step is called verification.
   b) A **Build** is ran only for the sources that are chosen in the verification step.
   c) A summary of the updated sources is displayed in the **Message** pane.
5. To update files from mainframe library: to make sure that you have the current version of the resources that are brought to your project from mainframe use **Update Modified Mainframe Members** function from the project menu or select **Update Modified Mainframe Members** from **Build** menu.
6. To build only the updated resources – Make: to make a build exclusively with the modified resources use **Make** option. Click **Make** from the toolbar, alternatively you can select **Make** from **Build** menu.

   Note: If you start a **Build** on a project where other users logged in, a warning message appears indicating which users are connected to the project. You need to confirm the operation.

   If another user activates a Build while you are logged in to a project a warning message appears urging you to close the project and wait for a notification that is sent to all users when the build process is completed. During the Build process, the project is locked and cannot be accessed by any user. After the Build is successfully completed, a notification is sent to all users logged in to the project.

   It is possible that the **Make** process might take a long time during which you cannot use the application. If you need to use the application, you can run the Make process in the background. To force the **Make** operation to run in the background, follow the steps:

   a. Click **Start**, select **Run** then type **cmd** followed by **ENTER** to open the command window.
   b. Go to the folder where your **IBM AD Build Client** is installed and locate `IBMApplicationDiscoveryBuildClient.exe` file. Drag the `IBMApplicationDiscoveryBuildClient.exe` file into the command window then enter **/?** and press **ENTER**. A window is displayed containing detailed instructions about how to make a specific

process to run in the background. To make the **Add files** operation to run in the background, follow the displayed steps.

Updating Projects

You can update a project in two ways: manually or automatically. The process of manually updating a project is described as follows. For details on the automatic process, see “V. Setting up Automatic Updates with Windows Scheduler” on page 305. If you want to update the project manually, this procedure takes only two steps from the project menu only.

1. Update Modified Mainframe Members.

   This action checks for all project members that originated from the mainframe, if a new version of their source is available.

   All sources that are brought from the mainframe have data about their mainframe origin and last update time, which is stored in the IBM AD repository for the project.

   For sources that were brought from Endevor, this action checks against Endevor if a new version for the file is available, since the last retrieval date. If a new version for the file is available, the member is brought to the mapped virtual folder that matches the Endevor library.

   For PDS members, **IBM AD Build Client** checks the file dates on the mainframe against the last update date from IBM AD repository. If the member on the mainframe is newer, it gets updated on the PC folder that matches the PDS name.
2. Make

   This action effectively updates the IBM AD repository with the information relevant to the modified sources, and keeps it up to date with the code in the sources on the mainframe.

   Make builds a small subset of the whole project, as an incremental build step after which the full project repository is up to date with the minimal effort needed.

   This action starts with checking all the project members on the PC disk folders against their last recorded update dates on the last build time that is stored in the IBM AD repository for the project.

   If a file on the disk is newer than the information recorded in the database, then the file is part of the **Project Make** process that is an incremental build. If the newly updated files are programs or jobs, then they are added to the list of components that must be added to the programs/Jobs to be built in the **Make** process.

   If the newly updated files are copybooks, then **IBM AD Build** checks in the repository for all programs that copy these files, and these programs are added to the programs to be built in the Make process.

   If the new updated files are JCL PROCs, or JCL Include Files, or JCL Control files (PARMLIB files) then **IBM AD Build** checks in the repository for all JCL Jobs that use these files, and these Jobs are added to the programs to be built in the **Make** process.

   After this stage, **IBM AD Build** runs a build for the programs and Jobs that must be updated according to the previous steps, and after these components are built a summary of the number of updated components appears on the **Make** log.

   The `Make` log, just like any `Build` log, is saved to the disk under the project folder, with the **Make** date time. This method allows viewing past **Make** results and updated components at any time.

   You do not need to do anything on **IBM AD Build Configuration** for this update of Endevor and PDS members.

   For the CA7 manual update, the way to start the CA7 Data retrieval is by using the **IBM AD Build Configuration**, by using **Querry Environment** > **CA-7 Workload Automation** option.

Synchronizing Mainframe Members

The **Synchronize Mainframe Members** feature allows the user to specify whether **IBM AD Build** must update against specific libraries, where to add/remove the related members in/from the project (that

is, which virtual folder to use) and also which type of members **IBM AD Build** must use when you add members. The basic assumption is that the specified libraries do not contain members that do not need to be added even though they are there.

The **Synchronize Members** action can be performed against sources that are based on the **Host Side** and also locally on the **PC Side** (local/remote path).

The **Synchronize Members** action is run by using a configuration file that specifies what members of what type to be brought into which mapped virtual folder of the project. When you run **Synchronize Members** on a project, only the members that belong to libraries specified in the configuration file for this particular project is synchronized.

Note: When adding mainframe members, if the limit of **40000** members is reached, additional virtual folders are created at the project's level. They do not have correspondent on disk and they cannot be deleted manually from **IBM AD Build Client**. The additional virtual folders are different than the existing virtual folders by having a special naming convention, suffixed by **@&@<incremental_number>**. Examples of additional virtual folders for `zOS Cobol`:

```
zOS Cobol@&@1
```

```
zOS Cobol@&@2
```

The **Synchronize Members** feature is activated from **IBM AD Configuration Server**.

For details on the syntax of the configuration file and an example, see “Synchronize Members Configuration File Examples” on page 494.

After the members’ synchronization process is finished, use **Make** to ensure that the analyses you ran are done on the current version of the mainframe sources (updated, added, or removed).

Displaying Build Results

On the completion of the build process, you can view:

- The information that was collected and stored in the IBM AD Build Client project repository. For more information, see “Analyzing a Project” on page 456.
- A build error log and a series of reports that are generated that are generated under the **project's folder** > **reports** folder.
  - The `BuildErrorReport_<timestamp>.html` build error log that contains all the errors that were encountered during the build process.
  - In the **Check** folder you can find reports that are related to the missing components.

    Example of `.csv` file.

    ```
    MissingComponentsReport_<timestamp>.html
    ```
  - In the **Resources not saved in the repository** folder you can find reports that are related to **jobs** and **programs** that were not saved in the repository.

    Examples of `.csv` files.

    ```
    Jobs not saved in the repository_<timestamp>.csv
    ```

    ```
    Programs not saved in the repository_<timestamp>.csv
    ```
  - In the **GenericAPI** folder you can find reports that are related to crypto CALLs services:

    Example of `.csv` file.

    ```
    GenericAPI_<timestamp>.csv
    ```

Example of `.html` file.

```
GenericAPI_<timestamp>.html
```

For more information, see Interpreting the report files.

– In the **Unresolved Dynamic Calls** folder you can find reports that are related to all the dynamic calls that were not resolved by the compiler, during the build process.

  Example of `.csv` file.

  ```
  ProgramCall_Resolutions_<timestamp>.csv
  ```

  Example of `.html` file.

  ```
  ProgramCall_Resolutions_<timestamp>.html
  ```

  For more information, see "Interpreting and Handling the Unresolved Dynamic Calls report" on page 268.

– The **main build report** that includes all the above-mentioned reports. A message is shown in the output window, informing that the main build report can be found at a specific location. You can also double-click this message to open the `.html` report in your default browser.

  Example of main build report.

  ```
  MainBuildReport_<timestamp>.html
  ```

Interpreting and Handling the Unresolved Dynamic Calls report

The `ProgramCall_Resolutions_<timestamp>.csv` file contains six columns:

Important: Under any circumstances do not change the structure of the `.csv` file.

- The first column contains the Path of the file where the call is performed.
- The second column contains the Path of the Program that is using, calling, or invoking the file where the call is performed.
- The third column contains the row where the call is found.
- The fourth column contains a static value that **must not** be changed.
- The fifth column contains the unresolved dynamic calls that must be changed with the desired value.

  Note: Note that this is the only column that needs to be modified.
- The sixth column contains a static value that **must not** be changed.

The following procedure helps you to manage all the unresolved dynamic calls.

1. Open the `ProgramCall_Resolutions_<timestamp>.csv` file.
2. Update the `.csv` file with the right values for the unresolved dynamic calls.
3. Once all the calls have been resolved, rename the `ProgramCall_Resolutions_<timestamp>.csv` file to `ProgramCall_Resolutions.csv` and copy it in the **project's folder**.
4. Build all the programs that were affected by the changes.

   Note: In order to build only the affected programs, copy all the paths from the second column present in the `.csv` file, go to IBM AD Build Client, right click the project name, choose **Build Selection...**, and paste all the paths in the main text area of the **File List** window.

Working with IBM AD Build Client windows

A number of special purpose windows facilitate user interaction. Some of these windows are initially docked to the borders of the display area, but they can be undocked and moved, resized, and hidden (closed). Window names are not shown on docked windows.

Many of the windows have menus that are opened by placing the mouse cursor over the window and right-clicking. In some cases, different menus appear, depending on the exact position of the cursor in the window.

The following IBM AD Build Client windows are described in the following sections:

- Decisions
- Editor
- Settings
- Properties.

Viewing Source Programs

To view the source code for a particular entity, follow these steps:

1. Right-click the entity to open a menu, as described in "Right Click / Shortcut Menus" on page 245.
2. Select **View Source**. An **Editor** window opens containing the source listing of the entity.

Building Decisions

IBM AD Build Client's Decisions mechanism lets you to overcome syntax problems that might occur at build time in some source dialects. No permanent changes are made to the original code files. Instead, the change information is stored in the repository, so that when an analysis process requires a source file, in effect a temporary internal copy of the file with the modifications is used.

Decisions are essentially specifications for **Find** and **Replace** operations that can be applied locally (at a particular location in a specified file) or globally (throughout the application). This method allows for increased flexibility as decisions can be targeted to specific files.

The folder types that are accepted by the Decisions mechanism are as follows:

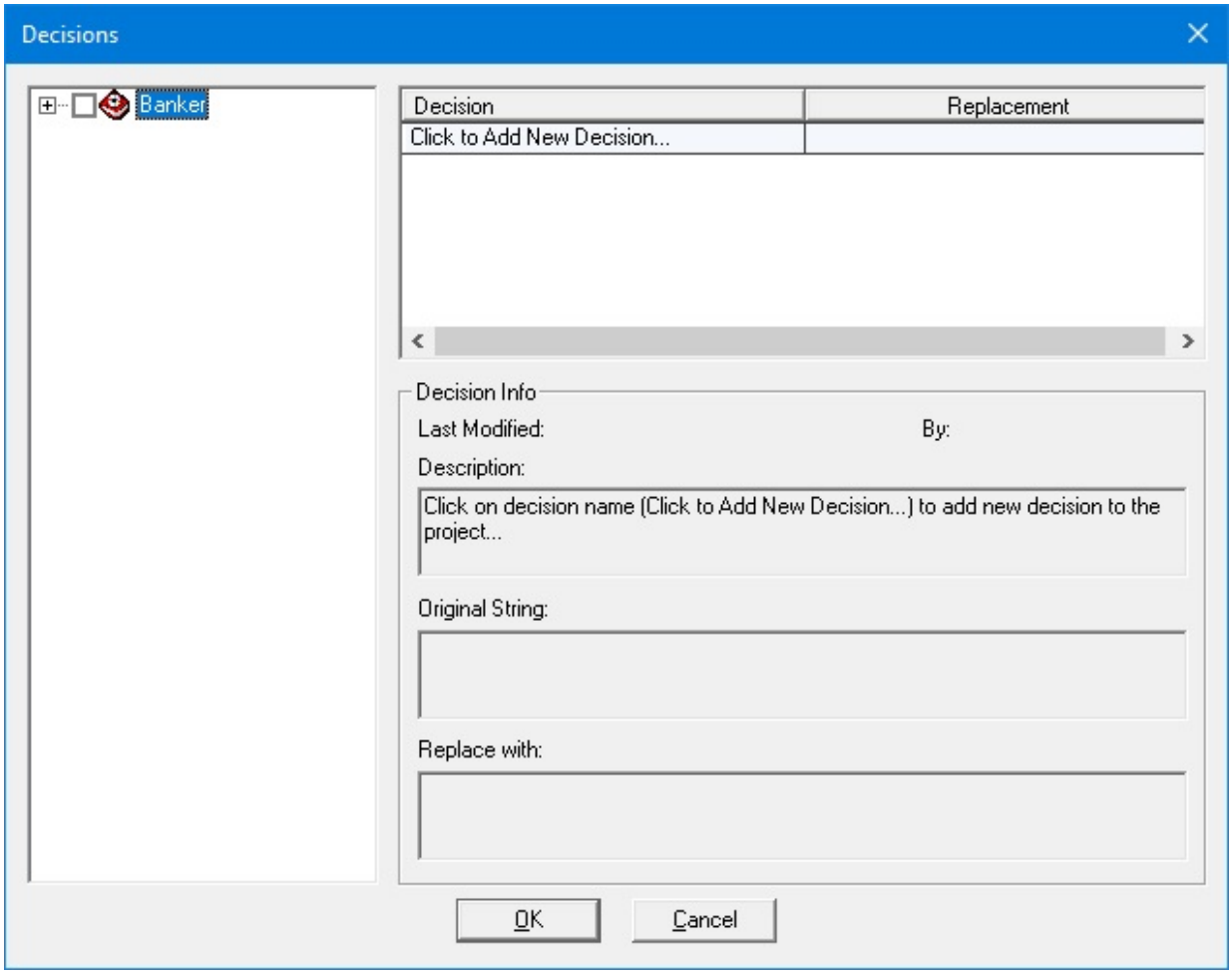
- zOS Cobol
- Cobol IDMS
- DT Cobol Pre-compiled
- Cobol IDMS Record

A decision might be implemented for any number of reasons, for example:

- The effects of *modifying* transactions can be studied.
- Build errors can be corrected.
- Unsupported COBOL features can be replaced by alternative code.

To define a decision, follow these steps:

1. Click **Build** / **Decisions** to open the **Decisions** window. If decisions are defined previously, they are listed in the window, otherwise the window is empty.

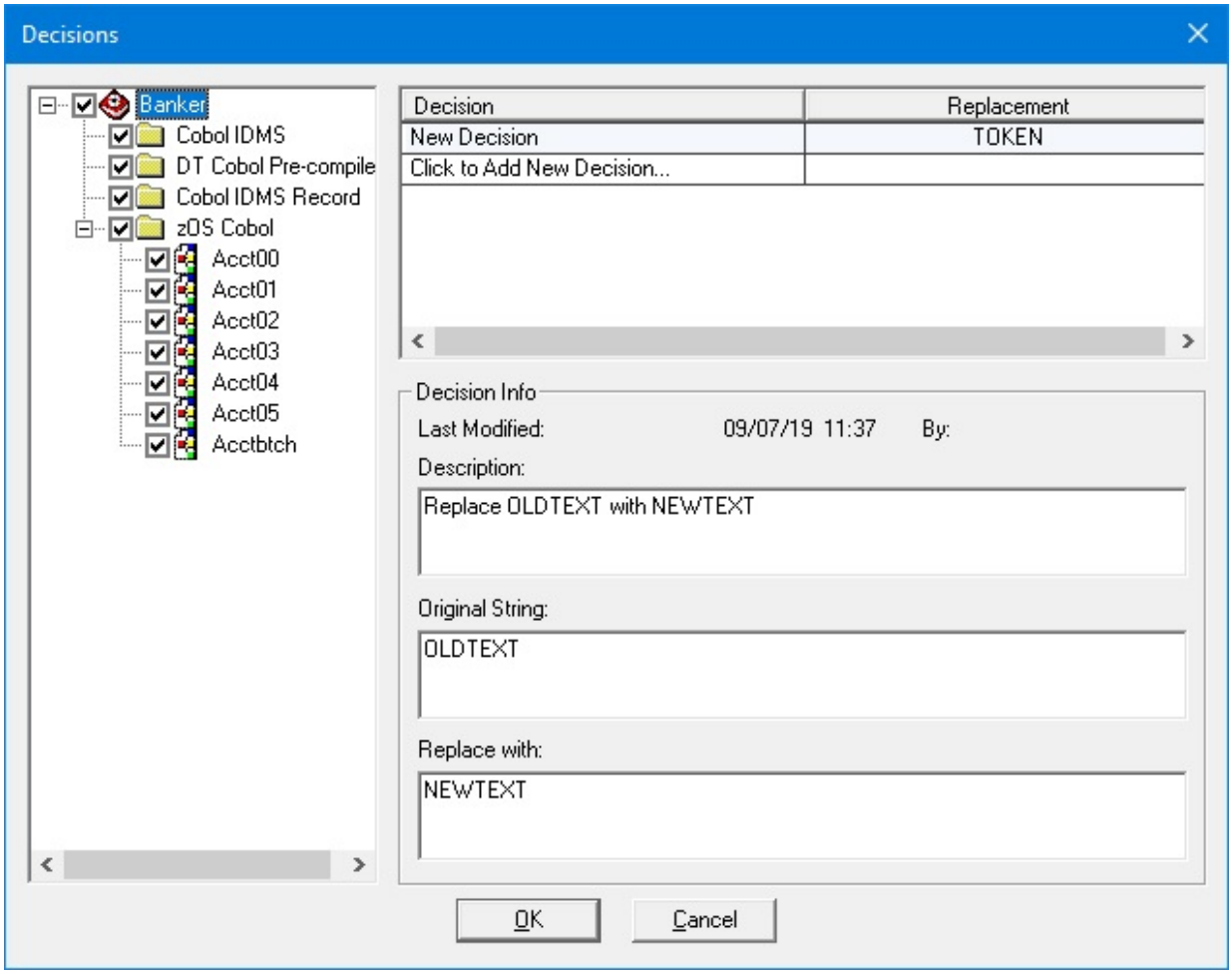


2. In the **Decision** pane at the upper right part of the window, click the text **Click to Add New Decision** (the **Decision** column) to create a new decision. This pane contains two columns and a row for each decision. When you click **Click to Add New Decision**, a new row is added for the new decision. Overwrite **Click to Add New Decision** with a name for the decision, and then click in the Replacement field and select **TOKEN** or **PATTERN** from the list menu box.

   The replacement type refers to the method that is used for search and replace operations. In this aspect, **IBM AD Build Client** follows the rules of **COPY REPLACING statement** from the COBOL language. In **PATTERN** search, the search string is replaced wherever it appears, while for **TOKEN** only complete words are replaced. For example, if the string **OLD TEXT** is to be replaced by the string **SOME NEW TEXT** using **PATTERN** search, a part of the string **BOLD TEXT** would be replaced by **SOME NEW TEXT** resulting in **BSOME NEW TEXT**.

   For **TOKEN** searches, the string to be replaced must contain one word only and only complete words are replaced. Thus, if you choose **TOKEN** replacement and if you want to replace **OLD TEXT** with the **NEW TEXT**, in the source file that contains the phrase **BOLD TEXT** the replacement would not take place although **OLD TEXT** is the suffix of **BOLD TEXT**. In this situation, the string **OLD TEXT** would be ignored, since it comprises two tokens.

   After the replacement mode is selected, a tree diagram of the project will appear in the left pane of the window (see the next image). If necessary, expand the tree.



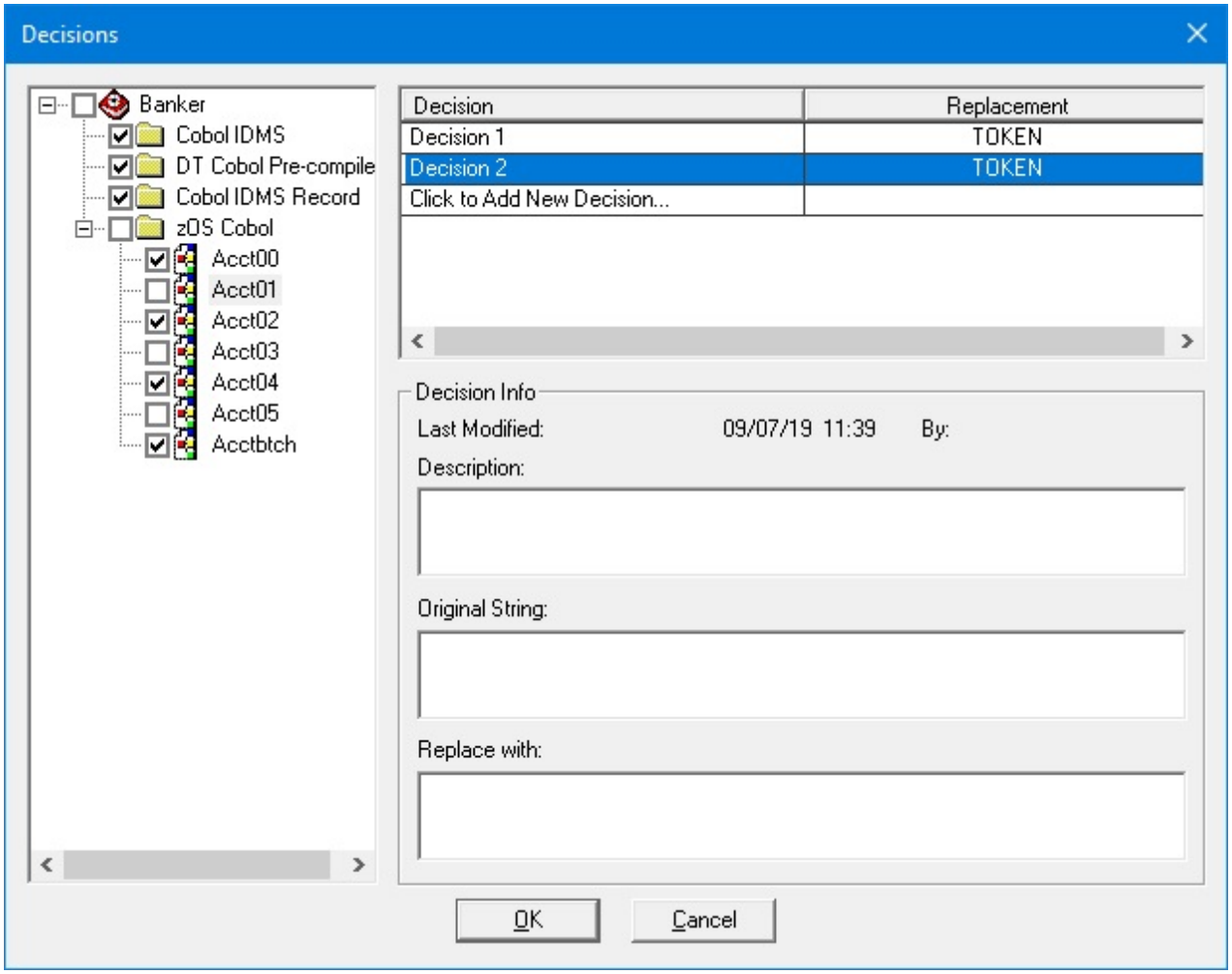
3. Enter a description of the decision in the **Description** text entry box, the string (or token) to be replaced in the **Original String** box, and the replacement string in the Replace with box.
4. Expand the project tree in the left pane to show its folders and files. Set the check boxes of the files that are to be included in the Search and replace operation. Some folders do not support decisions and therefore, their respective check boxes are disabled. (In the example that is shown before, the replacements are to be made in all the sources).
5. Click the check boxes next to the file name that is to accept the decision. A check mark appears. The decision is now attached to the *checked* file. Repeat for all the files or folders to which the decision is to be attached.

Additional Decisions

Each decision is represented by a row in the **Decision** pane, and have its own description, original string, replacement string, and program tree that specifies the files to which the decision is to be applied.

Information including the date of the last modification to the decision and the user name of the person who made the modification is displayed.

When you click a row to select it, the information in the other controls of the window changes. The following figure shows the decision information for two decisions.



Deleting a Decision

After you define the decisions, they remain active until they are deleted or until all check boxes in the project tree for the decision are cleared. In other words, if a decision is not associated with any files (all check boxes in the project tree for that decision are cleared), then the replacement it defines is not implemented, but the decision is still available for later use. To permanently delete a decision, select it and click **DELETE** on your keyboard.

Applying Decisions

After you define or modify decisions, the project (or the files that are affected by the decisions) must be rebuilt.

Note: A source opened in the text editor does not show applied decisions, since these decisions are applied only at build time on a temporary copy of the source code.

Using the Editor

IBM AD Build Client includes an integrated text editor that can be used to view files.

To open a file in the **Editor**, follow these steps:

1. **Main Menu** > **File/Open** to open the standard **Windows Open File** dialog box. Any file can be opened in this way, including files unrelated to **IBM AD Build Client** activities.
2. Double-Click **Source File** Icons. In the tree diagram of the **Project** pane, double-clicking an icon that represents a source code file or statement causes the corresponding file to open in an **Editor** window, often with the appropriate statement highlighted.

3. **Shortcut Menus**. Most menusmenus that are associated with program/statement icons in the **Project** pane have a **View Source** option.
4. • Double-Clicking **Compilation Error** messages in the **Output** Pane. If errors occur during a build, they are listed in the **Output** pane. Double-clicking the error notification causes the corresponding source file to be opened in an editor window with the erroneous statement highlighted.

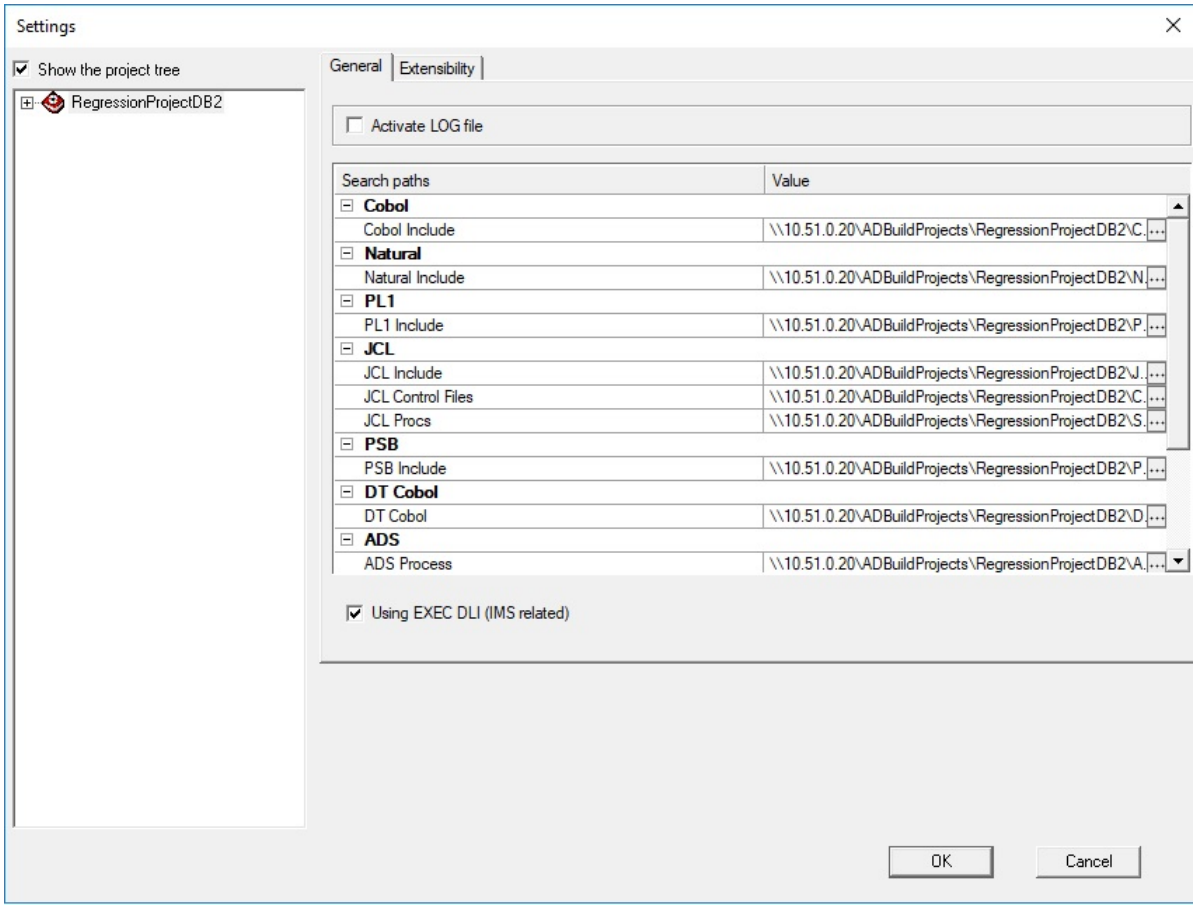
Using the Settings Option

Use the **Settings** option to change the default search paths that are used for the build operation to exclude program components from the build analysis, to set custom component extensions, and to select different analysis parameters according to the resource type.

To open the **Settings** window for a resource, follow these steps:

1. To open the **Settings** window for a resource, follow these steps:
   a. On the **Project tree** diagram, right-click the component that you want to exclude from the analysis.
   b. In the menu that opens, select **Settings**. In the **Settings** window select **Show** the project tree. The project tree is displayed showing the selected resource (you can select several resources if needed).
   c. If you select **Exclude File(s)** from **Build**, the selected files are excluded from subsequent builds. The parameter can be set for any individual file in the project, for a set of files within a folder or for an entire project folder, by selecting the folder node in the tree.

   Note: The options that are shown in the **Settings** window depend on the type of resource selected.
2. To open the **Settings** window for a project, follow these steps:
   a. Select the **Project** node in the **Project** pane.
   b. Go to **Project**, then select **Settings** to display the **Settings** window as in the following image.



In the **General** tab of the **Settings** window, after the **Show the project tree** check box is selected, the following check boxes, fields, or options are available:

Project / Folder	Check Box / Field / Option	What it does
Project node	Activate LOG file	Creates a log file of errors/warnings. This check box is password that is protected for administrator use only.
	Search Paths area	Displays the default search paths that are used during the **Build** operation. Allows the user to change the default search paths if needed. When you click **Explore**, the **Search Paths Order** window is displayed. Use the available buttons to either create an entry, to delete the selected entry, or to change the position of the selected entry in the list. The build operation is run in the order set in this window.
All folder nodes, individual resources	Exclude files from build	Excludes the select files or folder from build operation.
Natural	Indent size	Determines the column number where the text must start (in the source code).
Natural	Compiler Mode- Structured mode/ Report mode	Sets either the **Structured** or the **Report mode** for the compiler.
All include folders	(Cobol, Natural) Include Extensions or Default extensions	Allows the user to enter custom include extensions or use the default ones.
Data Area (in Natural projects)	Data Area file format- Format 1 or Format 2	Allows the user to select either **Format 1** or **Format 2** for the Data Area resources' processing. The option that is selected by default is **Format 2**.
PL1	PL1 Line Settings- Line Offset, Free Text format.	Allows the user to select the column number where the text must start (in the source code) or choose the **Free Text Format** option.

From the **Settings** window, an **IMS DB Environment** can be set up for COBOL programs that use **EXEC DLI** commands and **DL/I** calls. Fore more information, see step 4 from Adjusting Settings section.

From the **Settings** window, select the **Using EXEC DLI (IMS related)** check box to analyze COBOL programs with **EXEC DLI** commands that are present in the project. Fore more information, see step 5 from Adjusting Settings section.

The following check boxes are available in the **Extensibility** tab from the **Settings** window:

- **Enable API/Macro handling by using a configuration file.**
- **Enable handling of before and after preprocessed source code**.

The Options Window

This function from the **View** menu opens **Options** dialog box where you can specify the output parameters.

Specify the maximum number of output lines, whether warning messages must be displayed in the **Output** pane and if the build results must be automatically saved then click **OK** to apply the options.

The Properties Window

The **Properties** window displays information about the files in a project. The window can be opened from the menu of the items in the tree of the **Project** pane only. The labels and title of the window differ slightly according to the object type.

CICS CSD Information Handling

Online programs that run under CICS require access to external data sources, such as files, tables, and queues, cannot rely on jobs to perform the mapping to physical data source entities. CICS provides a way to define such mapping and saves the mapping information in the CICS System Definition (CSD) file.

To obtain CICS CSD information, the user can choose either of the following two methods:

- Using **IBM AD Connect for Mainframe**
- Using an exported CSD report

It is recommended to choose only one method to obtain CSD information in a project. For example, if the CSD information is obtained by using **IBM AD Connect for Mainframe**, and afterward the user decides to use an exported CSD report, the previous CSD information is automatically deleted. For more information, see “Deleting data from the repository” on page 276.

Using IBM AD Connect for Mainframe

When **IBM AD Connect for Mainframe** is used, the obtained CSD information is stored in the following MFCICS tables: MFCICSFile, MFCICSGroup, MFCICSGroupVsEntity, MFCICSGroupVsEntityLinks, MFCICSInfo, MFCICSInfoFiles, MFCICSList, MFCICSListVsGroup, MFCICSMap, MFCICSProgram, MFCICSTransaction, and MFCICSTransactionPerformance.

The information from the MFCICS tables, of the related database, is shown in graphs, reports, and usages in **IBM AD Analyze**.

Using an exported CSD report

When a CICS administrator wants to use an exported CSD report, a CICS utility, called DFHCSDUP, is used to extract information out of CSD. The result is a report that is generated by the **LIST** command of the DFHCSDUP utility.

The CICS administrator needs to carefully decide which parameters are used when the DFHCSDUP utility is invoked. The format is as follows:

```
         .-All--------------------.
>>-LIst--+----------------------+--+---------+---------------><
         +-Group--(--groupname--)-+  +-Objects-+
         '-LIst--(--listname--)---'  '-Sigsumm-'
```

In some cases, a CICS application uses a specific **LIST**. It is recommended to have a single application in a specific project and to use the appropriate list name when you run the utility. For example, **LIST LIST(listname)** is preferred instead of **LIST ALL OBJECTS**.

When the user specifies **LIST ALL**, the CSD report is parsed to save all the lists in the repository. In this case conflicts can occur. For more information, see “Conflict resolutions” on page 276.

The report is added to an AD project as a CSD type of file. The build process parses the file and stores the information in the MFCICS tables.

CSD report parser

The CSD report parser collects the following information:

- CICS region name.

  When an exported CSD report is used to obtain CSD information, the CICS region name represents the name of the report file. The region name is specified in the CICSName column of the MFCICSInfo table.

- The list of the CICS LIST components.
- The list of the CICS GROUP components and their relationship to the parent LIST.
- The list of the following CICS items:
  - Files
  - Map sets
  - Programs
  - Transactions

Important:

- If a group is not related to any list, it means that the group is not included in the group lists, specified by the CICS system initialization parameter **GRPLIST**, that CICS installs at cold start. This **Orphan Group** is excluded from the parsing and the resources contained by this group are not saved in the repository.
- The parser saves the first mapping relation encountered and ignores the others, when transaction is mapped to multiple programs.

After parsing the CSD report, the following information is used in **IBM AD Analyze**:

- The mapping between transaction and programs. The values are stored in the MFCICSTransaction table.

  Important: Only programs referred (that exists or are used) in the current **IBM AD Build Client** project are taken into account, in the mapping relation.

  The mapping is used/visible in analysis like:

  - **Program/Transaction Callgraph**
  - **Program Flow**
  - **Backward/Forward Call Chains** reports
  - **Explore project** as CICS Transaction, Resource Type
- The mapping between CICS files and their related dataset names, similar as dataset mapping in batch applications. The values are stored in the MFCICSFile table.

  Important: Only files used in existing programs in the current **IBM AD Build Client**project are taken into account, in the mapping relation.

  The mapping is used/visible in analysis like:

  - **Dataset Record Structure** report
  - **Dataset Usage in Programs**
  - **Explore project** as dataset, Resource Type

Deleting data from the repository

Deleting information from a previous region when querying a new one ensures the repository that has information from a single region each time. Whenever an exported CSD report or **IBM AD Connect for Mainframe** is used to import data all previous data from the repository is deleted. The user needs to consider that:

- All lists, inside the report, are considered for the name resolution.
- The region name is the CSD report file name.

Conflict resolutions

Multiple lists

There is the case when a transaction name is mapped to a program, in one list, and to another program, in another list. The CSD parser saves the information that is found in the first list and ignores the other mappings, from the other lists.

Multiple regions

In the context of an application analysis, do not store information from multiple regions in the same repository. The user is advised not to use more than one CSD report per project. It is not recommended to use the CSD report and retrieving operation information from **IBM AD Connect for Mainframe** in the same project. The **IBM AD Build Client** keeps the information from a single CICS region, based on the last **IBM AD Connect for Mainframe** action or CSD report parsing. The region is imported either by **IBM AD Connect for Mainframe** or by the CSD report.

Extensibility

Preprocessing Extensibility

In-house support preprocessors allow customers to view their familiar source code before preprocessing, while having **AD** parse the unfamiliar source code after preprocessing. Language preprocessors (also known as precompilers) are used to convert non-standard COBOL (for example) or non-COBOL code embedded in COBOL, into a form that the compiler can process. A non-integrated preprocessor takes as input a source file (defined as *before files*) reads and parses it then produces a modified source file (defined as *after files*) which is then passed as input to the COBOL compiler.

Note: The preprocessing extensibility feature allows IBM AD users that have their own COBOL preprocessor to see in the AD analysis the unprocessed sources.

IBM AD Build Client can analyze COBOL, PL/I and ASM applications that use preprocessors.

In order to access the **Preprocessing Extensibility** feature, there is an option on the interface after the project creation, named **Enable handling of before and after preprocessed source code** that will create the following required folders.

- New folder for *before files* named `PreProc Before`.
- New folder for *metadata* files named `PreProc MetaData`.
- New folder for *config* files named `PreProc Config`.

To enable the **Preprocessing** feature, **right click on folder tree** > **select Settings** > **click Extensibility tab**. For more details, see “Adjusting Settings” on page 253.

The folders are added as an option after the project is created, so those users not using preprocessing will not get confused.

Additional to the files above, *after files* must also be added. The *after files* will be added in the folder corresponding to their type (such as Cobol, PL/I, Assembler).

Before Files

These files represent the user's original resources.

Metadata Files

The metadata files map the *before files* with *after files*. These files must have the same name as the files to be compiled and the extension specified in the configuration file. The metadata file will have a `JSON` format. For details on the syntax of the JSON file and an example, please see “Preprocessing Extensibility Examples” on page 497.

The following elements from the JSON file, are explained below:

info
Contains information about the **json** format.

version
Version of the format.

metadata
An array that contains metadata elements for the before/after file pair.

pathType
Specifies whether the *before*, *after*, and *copybook* paths are set in mainframe format or local PC/network paths. Valid values for this attribute are *MF* (for mainframe path format) and *PC* (for local/network path format).

beforePath
Path to the original file, before the preprocessing process.

afterPath
Path to the expanded file, after the preprocessing process.

Tip: The **beforePath** and **afterPath** can be specified either in local/network path format or in mainframe format. For the mainframe format, only the PDS format is supported: *libray_name* (member_name). If the users use Changeman or Endevor to retrieve the sources, local network paths are required to be specified in the *beforePath* and *afterPath* values.

diffResolution
An array containing lines/columns mappings between the **beforePath** and **afterPath**, mapping established by the preprocessor.

beforePos
The corresponding position in the original file.

afterPos
The corresponding position in the expanded file. A position is defined with the following attributes:

- **startLine** - start line of the position.
- **endLine** - end line of the position.

If lines from a copybook exist in an after file, the following elements must be added in the corresponding metadata file:

type
Specifies if the lines in the after file are from a copybook. This element is required only when the lines in the after file are from a copybook; the only supported value is `INCLUDE`.

path
Specifies the path of the copybook that the lines come from. It can be specified in the local path or mainframe format.

includeStmtPos
Contains the following two elements that specify the include command position.

includeStmtPath
Specifies the path of the before file that includes the copybook.

includeStmtLine
Specifies the line number of the include command in the before file.

Note: The file will be added in the `PreProc MetaData` separate virtual folder under the project, so it can be updated from the mainframe if required. For more details about metadata files, see section Extensibility preprocessing JSON schema in appendix 3.

Metadata Files - Error Cases Behavior

The format and content validation will be performed at build start on the corresponding *after file*.

- If the JSON validation fails, the build on the file is stopped and the data in the database is cleared.
- If the *beforePath* or *afterPath* values in the JSON file do not exist on the disk, the build on the file is stopped and the data in the database is cleared.

Configuration Files

The configuration file will contain mappings between the folders of the *before*, *meta* and *after* files and the extensions for each type. When compiling a file from folder X, a search is initiated for a metadata file in the meta folder corresponding to folder X. The metadata file must have the same name as the file being compiled and the extension specified in the configuration file.

Important:

1. Each ***before*** folder must be associated with a unique ***after*** folder and can only have a **1 - 1 relationship** with each other. The same ***after*** folder **cannot** be used for multiple ***before*** folders.
2. The file will be added in the `PreProc Config` separate virtual folder under the project, so it can be updated from the mainframe if required.
3. The paths in the configuration file must be specified in local/network format, not mainframe format.
4. Lines in the configuration file can be commented by adding ***** at line start.

For details on the syntax of the configuration file and examples, please see "Preprocessing Extensibility Examples" on page 497.

Configuration Files - Error Cases Behavior

- *Configuration file format related errors.*

  The configuration file format will be validated at build start:

  1. If the configuration file format is incorrect, the build will not be started nor will affect the data in the database.
  2. If several files of different types are built in the same session (PL/I, Assembler, Cobol and JCL) while the configuration file format is incorrect, the build for JCL will not be affected.
  3. If several configuration files are used, out of which some are incorrect, the behavior is similar to case 1 and error messages will be generated for each incorrect configuration file.
- *Other types of errors.*

  1. If the *after file* is present in the *after folder* while the *meta files* and *before files* are missing from their specific folders (*meta folders*, *before folders*), an error is logged without saving anything in the database about the *after file*.
  2. If the *after file* and *before file* are present in the their specific folders, while the *meta file* is missing from the *meta folder*, an error is logged without saving anything in the database about the *after file* and *before file*.
  3. If the *after file* and *meta file* are present in the their specific folders, while the *before file* is missing from the *before folder*, an error is logged without saving anything in the database about the *after file*.

Note:

1. If two types of resources (requiring / not requiring preprocessing) are available to a project, they must be organized in separate locations on the disk. In case this rule is not applied, the sources that do not require preprocessing will not be built.
2. The validation for the configuration file checks that the specified folders do exist on the disk.

Feature Known Behavior

If for a project, both *metadata file* and *configuration file* are used, the **Make Project** functionality will not be applied for these files. For more information about **Make Project** functionality, please see "Building Projects" on page 264.

API Call/Macro Extensibility

The **Extensibility - API** feature allows customers to access an analysis that reflects their usage of in-house or 3rd party APIs, by using a configuration file, instead of waiting for development support. Using JSON configuration files, the user describes how each API\Macro call is interpreted by IBM AD.

IBM AD Build Client supports API calls only for:

- COBOL and PL/I programs
- JCL jobs

For more information about the JCL jobs, see section "JCL Call Extensibility Examples" on page 513 in Appendix 4.

API call events can be handled as one of the following types of calls:

- Data access calls
- Inner application program calls
- Cross application calls

Important: The JCL call events can be handled only as inner application program calls.

The following statements are supported for API calls:

- `CALL PROGRAM`
- `EXEC CICS LINK PROGRAM`
- `EXEC CICS XCTL`
- `EXEC PGM` (supported only for JCL calls)

To enable the API macro extensibility feature, click **Project** > **Settings** > **Extensibility**, and then select the **Enable API/Macro handling by using a configuration file** check box. After you click **OK**, a folder with the name `API Config` is created.

In the `API Config` folder, three types of JSON configuration files can be added:

API Config
Specifies the API calls to be analyzed, the API call parameters, and for which one of these parameters, the values are needed.

User Exits Config
Contains a list of API calls and the path to a user exit.

The user exit is a JSON file or a utility that you must create. It contains new resolutions for the API calls. For more information about the user exit JSON files, see section "API/Macro Call Extensibility Examples " on page 500 in Appendix 4.

API Dependency
Allows the user to specify a new type of API extension that can be triggered regardless of any source code. By using dependency, the following mappings can be defined:

- a mapping between programs and generic transactions
- a mapping between programs and generic maps

For more information about the API Dependency, see section "Dependency Extensibility Examples" on page 518 in Appendix 4.

The resolution of the API calls is made by using a module that is called the **API Resolver**, which uses the `User Exists Config` JSON configuration file.

Note:

- After each compilation, a `JVME_Post_Compiler.log` file is created in directory `C:\Users\`*`User_Name`*`\AD\comp\log`.
- The `API Config` configuration file is validated before each build event and in case errors are found, an error message is displayed and the build stops.

The folder is added as an option after the project is created, so users that do not use the **API Macro** feature will not get confused.

Annotations

Starting with IBM AD V5.1.0 release, the API Resolver can add annotations on resolutions. The annotations are present in the `resolution.json` file. For more information, see "JCL Call Extensibility Examples" on page 513 and "Dependency Extensibility Examples" on page 518.

Note: Make sure that Annotations Database configurations from **IBM AD Configuration Server** are set.

Examples of annotations that are added by using the `"annText"` and `"annKeyword"` parameters:

- `"annText": "ANNOTATION2"` - specifies the text that users want to add as annotation.

- `"annKeyword" : "API_RESOLUTION"` - specifies the annotation keyword used by the user to identify specific annotations.

Error Cases Behavior

For any project that contains COBOL sources, before any **Build**, **Build Selection** or **Make** processes, the `configuration file` validation starts automatically. You can also manually start the validation sequence, by **right click JSON file** > **Validate**.

1. If the JSON file is valid, the following message is displayed:

   ```
   Validation of the configuration file from 'API Config' folder has succeeded.
   ```

2. If the user tries to upload more than one `API Config` type file, the following error message is displayed:

   ```
   Only one configuration file of each type can exist in "API Config" folder.
   ```

3. If the JSON file is not valid, an error message is displayed. The error message varies, depending on the error type:

   a. If an empty JSON file is added to the project, the error message is:

   ```
   Error parsing data for API Configuration file. Reason: The configuration file from 'API Config' folder is empty.
   ```

   b. If the JSON file has a syntactical error in its structure (ex: a missing bracket, an extra comma, and so on) the following error message is displayed :

   ```
   Error parsing data for API Configuration file. Reason: At line (line number), column (column number)
   ```

   Note: Depending on the syntactical error, **Reason** can be: **`not a value`**, **`not an array`**, **`not an object`**, **`not a pair`**, **`no colon in pair`**, **`not a string`**.

   c. If one of the keys of the JSON file or their values are incorrect, the following error message is displayed:

   ```
   Error parsing data for API Configuration file. Reason: Key '%s' is invalid or has invalid value.
   ```

   d. If any mandatory key is missing for the JSON file or its value is unsupported, the following error message is displayed:

   ```
   Error parsing data for API Configuration file. Reason: Key '%s' does not exist or has unsupported value.
   ```

   e. If the **Api/Macro** feature is enabled, but no JSON file is added to the project, the following error message is displayed:

   ```
   Error parsing data for API Configuration file. Reason: The configuration file from 'API Config' folder is missing.
   ```

   f. If the user sets same values for more than one **`apiKey`**, the following error message is displayed:

   ```
   Error parsing data for API Configuration file. Reason: The key has a duplicated value.
   ```

   Note: Same behavior occurs for setting same values for more than one **`Program "Name"`** or **`Parameters "label"`**.

   g. If an error that is not covered by the previously documented situations is encountered, the following default error message is displayed:

   ```
   Unknown error type.
   ```

Update API Resolutions usage using CLI

IBM AD Build Client can be invoked in batch mode to update **API Resolutions** by using the following command:

```
IBMApplicationDiscoveryBuildClient /uar1 ProjectName
```

Where:

- **/uar1** is the parameter that is used to invoke the update of API Resolutions.
- **ProjectName** is the name of the project where the API Resolutions process is triggered.

Note:

- It is mandatory to have the **API Extensibility** feature *enabled* in **IBM AD Build Client** . To *enable* the **API Extensibility** feature, go to **IBM AD Build Client** > **Project** > **Settings** > **Extensibility** > **Enable API/Macro handling by using a configuration file**.
- The **Update API Resolutions** must be used for the situation when the API Resolution File is modified, so the resolutions must be updated in the repository but without running a full build on the project (which might be time consuming).

The logs for the **Update API Resolutions** usage are available under: **Project's Folder** > **UpdateApiResolutions_timestamp.txt**.

The action that is performed in background, use cases and best practices are also available in HTML format.

1. Click **Start**, select **Run** then type **cmd** to open the command window.
2. Go to the folder where your **IBM AD Build Client** is installed and locate `IBMApplicationDiscoveryBuildClient.exe`.
3. Drag `IBMApplicationDiscoveryBuildClient.exe` into the command window then type **/?** and press **ENTER**.

As a result, a web page is displayed containing detailed information.

Configuring the PL/I Preprocessor

Make sure that **IBM® AD Build Client** is up and running, and a project is available and can be used.

When working with the PL/I Preprocessor, you can configure parsing options, which drive the way preprocessing is executed, environment variables, encoding, the default library, and other settings. The PL/I Preprocessor configuration options are specified in the `PL1PreprocessorInfo.ini` file. The file follows the general format of `.ini` files where options are specified as *key=value*. You can access and edit the file from **IBM® AD Build Client**.

1. Click **Project** > **Settings** and select **Show the project tree**.
2. Select **PL1** from the list and click **Edit Preprocessor Settings**.

   The `PL1PreprocessorInfo.ini` file is displayed in your default text editor.
3. Specify the configuration options to customize the PL/I preprocessor. See "PL/I Preprocessor Configuration File" on page 282 for details about the configuration options you can set and for an example of the PL/I Preprocessor configuration file.

PL/I Preprocessor Configuration File

`PL1PreprocessorInfo.ini` is the PL/I Preprocessor configuration file, which specifies parsing options, environment variables, and other settings that the user sets when working with the PL/I Preprocessor.

The configuration file is generated when a project is created, and is located in the `<ProjectRootDirectory>\ConfigurationExt\` folder.

A `PL1PreprocessorInfoDefaultSample.ini` file is available in the `<IBM AD Build Client installation folder>\Bin\Release\Samples\PL1_Preprocessor\` folder

The file follows the general format of `.ini` files where options are specified as *key=value*.

Sections and groups

Sections denote groups of options that override the options in the previous levels. Sections can be hierarchical, names of the groups must be separated by `/`.

Groups are the virtual folders that are created in the project's structure in **IBM® AD Build Client**.

Parsing Options

The parsing options drive the way preprocessing is executed. The parsing options are described as follows:

- `opts.blank.chars=<character set>`

  Specifies the characters that can be used by the preprocessor. By default, space, tab, newline are blank characters.

- `opts.margins=<true/false>`

  Specifies whether the files can have special margins. If the option is set to `true`, you must specify both the left and right margins.

- `opts.margins.left=<natural number>`

  Specifies the left side margin of the files to process, as a column number. Any text to the left of `opts.margins.left` is ignored. This option must be specified if `opts.margins=true`.

- `opts.margins.right=<natural number>`

  Specifies the right side margin of the files to process, as a column number. Any text to the right of `opts.margins.right` is ignored. This option must be specified if `opts.margins=true`.

- `opts.stringDelim=<character set>`

  Specifies the characters that can be used in text as string delimiters instead of the default `"`.

- `opts.or.chars=<character set>`

  Specifies the characters that can be used as the OR operator in preprocessor directives.

  Note: `opts.or.chars` is also used by the concatenation symbol. For example, if `!` is used as the OR operator, then concatenation symbol is `!!`.

- `opts.not.chars=<character set>`

  Specifies the characters that can be used as the NOT operator in preprocessor directives.

- `opts.extra.lower=<character set>`

  Specifies the extra **lowercase** characters that can be used in preprocessor identifiers.

- `opts.extra.upper=<character set>`

  Specifies the extra **uppercase** characters that can be used in preprocessor identifiers.

  Note: `opts.extra.lower` and `opts.extra.upper` must have the same length. Characters are matched based on their position.

- `opts.include=<non-spaced set of characters>`

  Specifies a custom include directive.

  Note: `<character set>` is a set of characters that are surrounded by any of the following pairs of separators: `{} () >< `` .. ~~ || ++ == __`.

- `opts.library.extensions=<comma-separated list of names>`

  Specifies the extensions of the PL/I includes that are used by the user. For example, if the includes have `.pli` or `.inc` extension, these mentioned extensions are written in the `opts.library.extensions` option.

- `opts.caseInsensitive=<true/false>`

  When the option is set to `true` (default), the compiler option `CASE(UPPER)` is implemented. When it is set to `false`, the compiler option `CASE(ASIS)` is implemented.

Ignore an include member

- `internal.ignored.member.files=MEM1;MEM2;MEM3`

  Specifies the members that will be ignored at the build time.

Note: After each member you must add the `;` (semicolon) separator, exception with the last member.

Other settings

- `source.encoding=<valid encoding>`

  Specifies the encoding that is used to parse files. The default is `UTF-8`. Valid encoding names are listed in List of supported encodings.

- `default.library=SYSLIB`

  Specifies the default library that the PL/I preprocessor looks for includes.

- `internal.include.flat.layout=<true/false>`

  When the option is set to `true`, it forces the preprocessor to ignore the library in an include directive.

Environment variables do not have a predefined value. Subsequently, a value can be assigned to one of these variables. A variable has the following format:

`vars.VARIABLENAME='value'`

Configuration file example

```
[PL1]
source.encoding=UTF-8
opts.include=++INC
opts.stringDelim={"}
opts.extra.lower={@\#$}
opts.extra.upper={@\#$}
opts.margins.right=72
opts.margins.left=1
opts.margins=true
vars.MODE=BATCH
internal.ignored.member.files=MEMBER1;MEMBER2;MEMBER3
[PL1/Subfolder1]
opts.include=--TST
opts.stringDelim={%}
vars.MODE=CICS
[PL1/Subfolder1/Subfolder2]
opts.margins.right=20
opts.margins.left=7
```

The example configuration file contains folders in hierarchy. If `[PL1/Subfolder1]` and `[PL1/Subfolder1/Subfolder2]` options are used, the actual options are compiled based on hierarchy, overwriting the parsing options present in `[PL1]`:

```
opts.include=--TST
opts.stringDelim={%}
vars.MODE=CICS
opts.margins.right=20
opts.margins.left=7
```

Configuring the Excluded Assembler Macros List

Important: The **Assembler Macro Expansion** feature is an extension of the base Assembler compiler and it is not enabled by default. To enable the **Assembler Macro Expansion**, see the "Enabling the Assembler Macro Expansion Feature" on page 238 section.

There are some special macros, also known as system macros, that are intensively used in most Assembler programs, but may not provide many useful insights. The Expansion Macro feature provides the possibility of not expanding or interpreting these types of macros by adding them into a excluded list of macros.

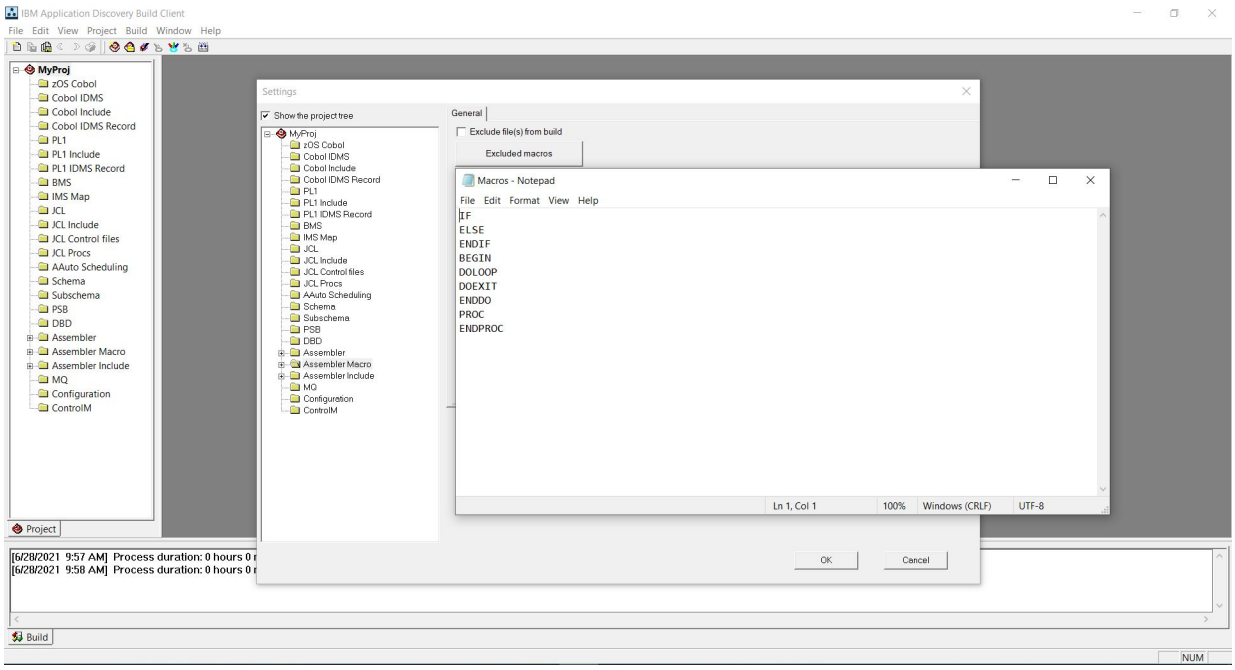
To edit the excluded list of macros, perform the following steps:

1. Make sure that **IBM® AD Build Client** is up and running, and a project is available and can be used.
2. Click **Project** > **Settings** and select **Show the project tree**.

3. Select **Assembler Macro** from the list and click **Excluded macros**.
4. The denied `Macros.txt` file is displayed in your default text editor.

   Note: By default, the denied `Macros.txt` file contains the following common macros.

```
IF
ELSE
ENDIF
BEGIN
DOLOOP
DOEXIT
ENDDO
PROC
ENDPROC
```



5. Add or remove any macros as needed.

Preparing repository using DDL scripts for Db2 on z/OS projects

Creating Db2 Database Using DDL Script

A Data Definition Language (DDL) script can be used to create a Db2 database. The `DB2Z_CreateObjects.sql` DDL script is located in the `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts` folder.

1. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts` and open `DB2Z_CreateObjects.sql` by using a text editor.
2. Locate and set the following parameters in the entire script.

   - `CREATE DATABASE <enter an appropriate name for the database>`
   - `SET CURRENT SCHEMA = 'enter an appropriate name for the schema'`
   - `SET CURRENT PATH = 'enter an appropriate name for the path'`
   - `SET CURRENT FUNCTION PATH = 'enter an appropriate name for the function path'`

   Note: The names of the database, schema, path, and function path must have a maximum length of 8 characters. Special characters cannot be used.
3. Run the script.

4. After you create a Db2 database and schema, you can attach it to a new created project. For more information, see “Creating a project” on page 247.

The desired Db2 database is created.

Deleting Db2 Database Using DDL Script

A Data Definition Language (DDL) script can be used to delete a Db2 database. The `DB2Z_DeleteObjects.sql` DDL script is located in the `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts` folder.

1. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts` and open `DB2Z_DeleteObjects.sql` by using a text editor.
2. Locate and set the following parameters in the entire script.
   - `SET CURRENT SCHEMA = 'enter the name of the schema'`
   - `SET CURRENT PATH = 'enter the name of the path'`
   - `SET CURRENT FUNCTION PATH = 'enter the name of the function path'`
3. Run the script.

The desired Db2 database is deleted.

Creating Annotations Database on Db2 on z/OS Using DDL Script

A Data Definition Language (DDL) script can be used to create **Annotations database**. The `DB2Z_CreateAnnotationDB.sql` DDL script is located in the `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts` folder.

1. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts` and open `DB2Z_CreateAnnotationDB.sql` by using a text editor.
2. Locate and set the following parameters in the entire script.
   - `CREATE DATABASE <enter an appropriate name for the database>`
   - `SET CURRENT SCHEMA = 'enter an appropriate name for the schema'`
   - `SET CURRENT PATH = 'enter an appropriate name for the path'`
   - `SET CURRENT FUNCTION PATH = 'enter an appropriate name for the function path'`

   Note: The default name of the database, schema, path, and function path is EZANNOT. The default name can be changed and can have a maximum length of 8 characters. Special characters cannot be used.
3. Run the script.
4. After you create the **Annotations database**, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Database names**. The **Database Names** settings page is displayed.
5. Enter the Annotations Database name and the schema name in **Annotations Database**. The Annotations Database is used to store the comments that are related to application’s lifecycle added by a user on a resource.

The desired **Annotations Database** is created.

Preparing repository using DDL scripts for Db2 for LUW projects

Creating Db2 for LUW Database Using DDL Script

A Data Definition Language (DDL) script can be used to create a Db2 for LUW database. The `DB2LUW_CreateDatabase.sql` DDL script is located on the machine where IBM AD Build Client

is installed, in the <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts folder.

1. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts and open DB2LUW_CreateDatabase.sql by using a text editor.
2. In the entire script, replace the default AA3C3F2F-9A66-4442-BF0A-A6FD18718F1F Global Unique ID (GUID) with your desired Db2 for LUW database name.

   Note: The database name must follow the general Db2 object naming rules.
3. Run the DB2LUW_CreateDatabase.sql script by using Db2 Command Line Processor (CLP).
4. Make sure that you are connected to the newly created Db2 for LUW database.
5. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts and run the DB2LUW_CreateObjects.sql script.
6. You can attach the created Db2 for LUW database to a new project. For more information, see "Creating a project with a Db2 for LUW (Linux, UNIX, and Windows) database" on page 251.

The desired Db2 for LUW database is created.

Creating Cross Applications Database on Db2 for LUW Using DDL Script

A Data Definition Language (DDL) script can be used to create a Cross Applications Database. The DB2LUW_CreateCrossDB.sql DDL script is located on the machine where IBM AD Build Client is installed, in the <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts folder.

1. Go to <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts and open DB2LUW_CreateCrossDB.sql by using a text editor.
2. In the entire script, replace the default AA3C3F2F-9A66-4442-BF0A-A6FD18718F1F Global Unique ID (GUID) with your desired Cross Applications Database name.

   Note: The database name must follow the general Db2 object naming rules.
3. Run the DB2LUW_CreateCrossDB.sql script by using Db2 Command Line Processor (CLP).
4. After you create the **Cross Applications database**, go to the machine where IBM AD Servers and Services are installed, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Database names**. The **Database Names** settings page is displayed.
5. Enter the Cross Applications Database name in **Cross Database** > **Name**.

The desired **Cross Applications Database** is created and configured.

Creating Annotations Database on Db2 for LUW Using DDL Script

A Data Definition Language (DDL) script can be used to create **Annotations database**. The DB2LUW_CreateAnnotationDB.sql DDL script is located on the machine where IBM AD Build Client is installed, in the <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts folder.

1. By default, when running the DB2LUW_CreateAnnotationDB.sql DDL script an annotations database with the default EZANNOT name is created.

   Note:

   - In case that you want to change the default EZANNOT name of the annotations database, go to <IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\DBScripts, open DB2LUW_CreateAnnotationDB.sql by using a text editor, and replace in the entire file the default EZANNOT name with the one that you desire.
   - The database name must follow the general Db2 object naming rules.

2. Run the `DB2LUW_CreateAnnotationDB.sql` script by using `Db2 Command Line Processor (CLP)`.
3. After you create the **Annotations database**, go to the machine where IBM AD Servers and Services are installed, access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments** > **"Your environment"** > **Servers and security** > **Database names**. The **Database Names** settings page is displayed.
4. Enter the Annotations Database name in **Annotations Database** > **Name**. The Annotations Database is used to store the comments that are related to application's lifecycle added by a user on a resource.

The desired **Annotations Database** is created and configured.

ChangeMan ZMF – IBM AD Validation Process

This feature is relevant only for ChangeMan ZMF users and has as prerequisites: IBM AD Validation Server and IBM AD Connect for Mainframe.

Introduction

IBM AD Validation Server is installed on a Windows machine, and it requires to have installed **ChangeMan ZMF** on the Mainframe. However, the CHANGEMAN PACKAGE processing does not require **IBM AD Validation Server** to be installed since **ChangeMan ZMF** information can still be retrieved for analysis without **IBM AD Validation Server**.

IBM AD Validation Server is required only if **ChangeMan ZMF** is installed and you need to automatically validate the COBOL source code's conformance to a certain set of coding rules, when staging a package in **ChangeMan ZMF**. **IBM AD Validation Server** receives notifications from **ChangeMan ZMF** to validate the provided COBOL source files, when **ChangeMan ZMF** is configured to notify the **IBM AD Mainframe Connect for Mainframe** agents.

In the context of **IBM AD Validation Server**, validation is the process of assessing the degree to which a COBOL source file conforms to a set of configured coding-rules, and then generates a `.pdf` report and then sends back to **ChangeMan ZMF** a return code together with any resulting messages that describe the error for the COBOL source file that was validated. If the **ChangeMan ZMF** administrator requires this automatic verification before staging COBOL sources, to prevent packages from being staged, it represents also a delayed staging and an initial effort to configure **ChangeMan ZMF**, **IBM AD Mainframe Connect for Mainframe** agents, **IBM AD Validation Server**, and a java reporting component that checks the coding-rules conformance and generates the `.pdf` report.

Currently there are `161` coding rules in the `CodingRuleIndex` DB table, with a self-explanatory **Description** column that can help you decide whether the configuration effort and eventually the delayed staging suits your needs.

How it works

To have this feature up and running, **IBM AD Validation Server** must be installed and configured. For more information, see "STEP 3. (Optional) Configuring IBM AD Validation Service" on page 64.

If only downloading or synchronizing the mainframe members from ChangeMan ZMF is needed, IBM AD Validation Server is not required. IBM AD Validation Server provides coding rule enforcement via synchronization with ChangeMan ZMF when a member is staged. Before a source file is staged, you can automatically validate the source code to a certain set of coding rules.

The validation process works as follows.

1. A member is compiled in ChangeMan ZMF (Cobol Program, Assembler Program for example).
2. IBM AD Validation Service receives an indication that a certain program, part of a package within an application is compiled.
3. IBM AD Validation Service triggers IBM AD Build Client in background mode for the following actions: **Synchronize** and **Build** selection.

4. The **Synchronization** process is described as follows:

   A ChangeMan ZMF validation request, after it is processed by IBM AD Connect for Mainframe, contains the program name to be validated and all its include files. It contains also the PDS library names and their corresponding members of the staged ChangeMan ZMF package. The required mainframe members are downloaded from the PDS libraries in batches and contain multiple validation requests.

   In the synchronization process, the `ProjectsMapping.txt` configuration file is used. However, the projects that are present in this configuration file must have a valid z/OS connection, with IP/hostname, and port number, that is attached and configured. These projects must be used exclusively for validation purposes. For more information about the `ProjectsMapping.txt` file, see Step 1, in "STEP 3. (Optional) Configuring IBM AD Validation Service" on page 64.

   Some projects can be used only for validation purposes and they are not supposed to be used in **IBM AD Analyze Client** analysis. It is recommended to add the project names in the `projects.blacklist=<Project>` parameter of the **IBM® AD Batch Server's** `project.properties` configuration file.

5. The **Build Selection** process is described as follows:

   Build Selection process is optimized in the validation context, by allowing multiple instances of **IBM AD Build Client** to be launched in parallel so that the validation of large batches of files will be faster. The degree of parallelism is configurable in the `ParallelValidationParameters.txt` configuration file. The project names that are used exclusively in the validation context for **Build Selection** must be entered in `ProjectsMappingParallelBuild.txt` configuration file. Based on the load and the two configuration files, several programs are built on a certain instance of **IBM AD Build Client**.

   To build the programs that are added to the project, the **include search paths** needs to contain the paths of the include files that are detected in the previous step.

   Note:

   - On a certain **IBM AD Build Client** instance the programs are built sequentially, while **IBM AD Build Client** is launched in parallel on different projects.
   - Before the build starts, the project's repository is cleared so these projects cannot be used for analysis. Compiled data that is added in the repository is only needed to generate a validation report on the built programs.
   - The **Build Selection** projects do not have the requirements of the *Validation projects*, but it is required to have the virtual folders from `FoldersMapping.txt` configuration file, as the programs are added in the virtual folders based on their types.

6. After the **Build Selection** process finishes, IBM AD Validation Service starts to generate Rules Based reports for the program that was previously staged. IBM AD Validation Service is configured to have different weights for the rules, each rule that is infringed has a value that is defined by the user in the IBM AD Validation Service configuration.
7. Return of the max weight value to ChangeMan ZMF. After the report is generated and the maximum weight value is calculated, it is returned to the IBM AD Connect for Mainframe that further passes this information as follows:

   - To ChangeMan ZMF in user option 0401.
   - In the user's terminal as a message (where user is the one that initially staged the Cobol Program in ChangeMan ZMF). The messages sent to the terminal can be configured in IBM AD Validation Server in the `CompletionCodeVsMessage.txt` configuration file (Refer to *IBM AD Installation and Configuration Guide* for details). In the situation when there's a weight that is not configured in the previous configuration file, then the user sees in the terminal the message `error in flow` and IBM AD Validation Server logs must be investigated for further details.

   The enforcement part is developed by the ChangeMan ZMF admin. ChangeMan ZMF can be configured, based on the returned code, to prevent a package with programs that are violating the coding rules from being staged.

8. Default max weight values and return codes that are currently supported by IBM AD Validation process and IBM AD Connect for Mainframe when you send the information to ChangeManIBM ZMF:

   - 0 - converted to VPAS and sent to ChangeMan ZMF in user option 0401.
   - 4 - converted to VWRG and sent to ChangeMan ZMF in user option 0401.
   - 8 – converted to VFAL and sent to ChangeMan ZMF in user option 0401.

   Any other values (except 99) – converted to NA and sent to ChangeMan ZMF in user option 0401.

   Return Code 99 – converted to DISS and sent to ChangeMan ZMF in user option 0401. This code is a special return code that is sent only for the situation when something went wrong in the Validation Process flow (such as synchronize failed, **Build Selection** that failed, or the report cannot be generated).

Enabling TLS Connection between IBM AD Validation Server and IBM AD Connect for Mainframe

For IBM AD V5.1.0.7 and earlier versions, the communication between IBM AD Validation Server and IBM AD Connect for Mainframe is unencrypted socket session. Beginning with version 5.1.0.8, the optional secure communication, which uses the Transport Layer Security (TLS) protocol, is supported by using the Application Transparent Transport Layer Security (AT-TLS) feature of IBM z/OS Communication Server.

1. Find the `TLSEncryption.text` file in the `<IBM AD Installation Folder>\bin\release\IBMApplicationDiscoveryValidationServer` folder, and then set the flag value to Y in the file.

   This configuration file contains a single flag byte. The flag can take the value Y or N. It is not case-sensitive. Y indicates that AT-TLS is enabled. If N is specified or the flag value is missing, the communication will not be encrypted. A sample file is provided in the `<IBM AD Installation Folder>\bin\release\IBMApplicationDiscoveryValidationServer\SampleConf` folder.
2. Generate a personal certificate that is signed by a certificate authority (CA) to represent IBM AD Validation Server, and the key to this certificate. After the CA certificate and key files are generated, store them in the `<IBM AD Installation Folder>\bin\release\IBMApplicationDiscoveryValidationServer` folder.

   This certificate must be in the PEM Based 64 format and contain string "—– BEGIN ..." on the first line. A self-signed certificate can be used, but this is not recommended for a production installation.

   Important: When creating the personal certificate that will be presented to the client, which is IBM AD Connect for Mainframe in this case, on behalf of IBM AD Validation Server, set the Common Name field equal to the numeric IP address of the Windows machine where the IBM AD Validation Server is running. Do not use the host name.

   You can create and manage digital certificates and their related key pairs in many ways. If you use OpenSSL, see the following command example:

   ```
   openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365 -nodes
   ```
3. Provide a copy of the CA certificate that is generated in the preceding step to the person who configures IBM AD Connect for Mainframe.

   Ensure that you preserve the selected format when the CA certificate is generated. In the preceding example, the format is CERTB64. The CERTB64 format creates a DER encoded X.509 certificate that is in the Base64 format.

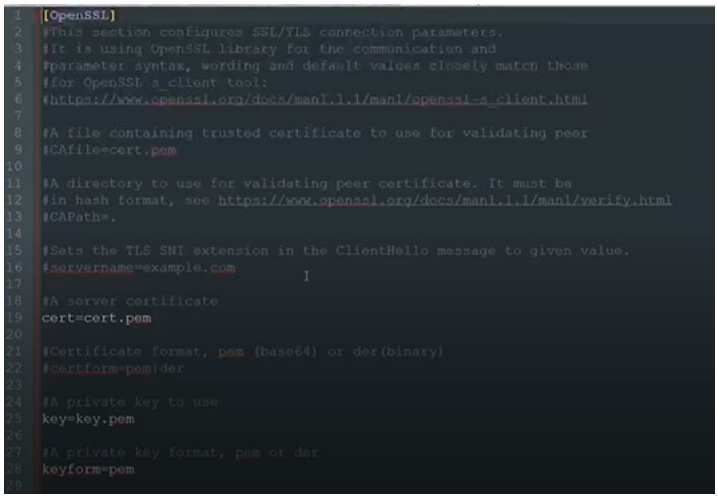
   During the TLS handshake process, the copy of the CA certificate will be used by IBM AD Connect for Mainframe to authenticate the personal certificate that is presented on behalf of IBM AD Validation Server.
4. Configure the `EZLCONN1.ini` file in the `<IBM AD Installation Folder>\bin\release\IBMApplicationDiscoveryValidationServer` folder.

This EZLCONN1.ini file follows the common INI file syntax. To enable TLS connection between IBM AD Validation Server and IBM AD Connect for Mainframe, the **cert**, **key**, and **keyform** parameters must be specified in this file.

cert=*cert*.pem
Specifies the file name of the CA certificate or the self-signed certificate that is obtained in step 2.

key=*key*.pem
keyform=pem|der
Specify the private key to the client certificate and the format of the key, which can be either PEM or DER.

Example



After setting the values of the **cert**, **key**, and **keyform** parameters, the EZLCONN1.ini file contains enough information to allow IBM AD Validation Server to connect with IBM AD Connect for Mainframe by using the TLS protocol. All the parameters must go under the OpenSSL section. The parameter syntax, wording, and default values closely match those for the s_client OpenSSL tool. For more information about the s_client OpenSSL tool and the supported parameters, go to https://www.openssl.org/docs/man1.1.1/man1/openssl-s_client.html. For more information about INI files, go to https://en.wikipedia.org/wiki/INI_file.

To enable TLS connection between IBM AD Validation Server and IBM AD Connection for Mainframe after you complete the setup for IBM AD Validation Server, the configurations for IBM AD Connect for Mainframe must also be completed. For instructions, see “Enabling TLS Connection to IBM AD Validation Server” on page 164.

Enabling TLS Connection between IBM AD Build Client and IBM AD Connect for Mainframe

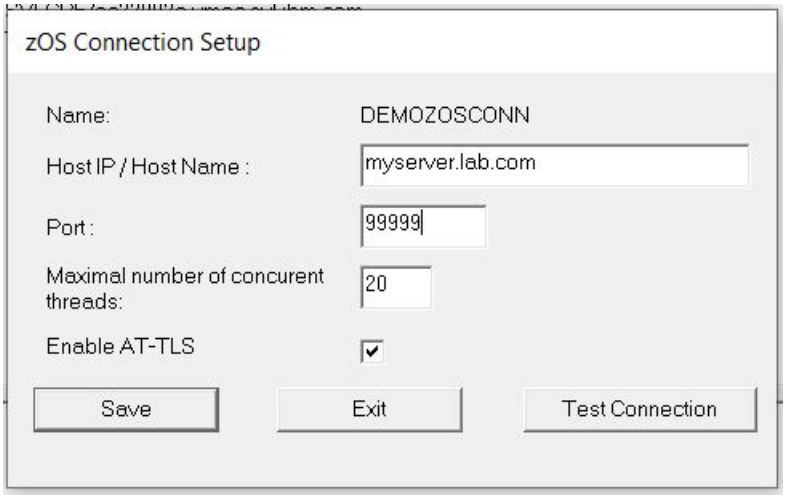
For IBM AD V5.1.0.4 and earlier versions, the communication between IBM AD Build and IBM AD Connect for Mainframe is unencrypted socket session. Beginning with version 5.1.0.5, the optional secure communication, which uses the Transport Layer Security (TLS) protocol, is supported by using the Application Transparent Transport Layer Security (AT-TLS) feature of IBM z/OS Communication Server.

The TLS protocol is a client or server cryptographic protocol. It is based on the earlier Secure Sockets Layer (SSL) specifications that are developed by Netscape Corporation for securing communications that use Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. The TLS and SSL protocols are designed to run at the application level. Therefore, typically, an application must be designed and coded to use TLS/SSL protection. OpenSSL is an open source implementation of the SSL and TLS protocols. It is widely used on most Unix-like platforms and Microsoft Windows systems. OpenSSL supports the full suite of SSL and TLS protocols (SSL V2, SSL V3, TLS V1.0, TLS V1.1, and TLS V1.2 as of this writing), including a robust set of application programming interfaces (APIs) and a wide range of cryptographic primitives.

IBM AD Build can establish a secure communication with the mainframe host by using the OpenSSL toolkit. OpenSSL provides a wide range of configuration settings to customize the connection. IBM AD Build Client supports some of these settings. You can specify the values in a `.ini` file that is located in the IBM AD Build Client installation folder.

On the host side, the communication is handled by using IBM AT-TLS services. Before you configure IBM AD Build to enable TLS connection, make sure to complete the AT-TLS setup for IBM AD Connect for Mainframe. For instructions, see "Enabling TLS Connection to IBM AD Build" on page 183.

1. Go to the **zOS** tab in IBM AD Build Configuration, and then create a z/OS connection with the **Enable AT-TLS** checkbox selected, or edit an existing connection to select this checkbox.



   For V5.1.0.6 and earlier versions, find the TLS file in the IBM AD Build Client `bin\release` folder, and then set the flag value to Y in the file. The full path is similar to `C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery Build Client\Bin\release`.

   The TLS file contains a single flag byte. The flag can take the value Y or N. It is not case-sensitive. Y indicates that TLS is enabled, while N indicates that TLS is not enabled. If this file is not present as a sample file, it can be created from the command line with command `echo Y > TLS` or `echo N > TLS`. There is no file extension on the TLS file name. The file must be saved in the IBM AD Build Client `bin\release` folder.

2. As part of the AT-TLS configuration process on the mainframe, a personal certificate that represents the IBM AD Connect for Mainframe server is created. This personal certificate must be signed with a certification authority (CA) certificate or a self-signed certificate. Save a copy of this CA certificate or self-signed certificate in the IBM AD Build Client `bin\release` folder. The full path is similar to `C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery Build Client\Bin\release`.

   The file that contains this certificate can have any name, but it must be in the Privacy-Enhanced Mail (PEM) format. If IBM AD Build Client must connect with multiple servers that use different CA certificates or self-signed certificates, this file can optionally contain multiple PEM format certificates. Standard practice is to place only trusted CA certificates or self-signed certificates in this file, but not the personal certificate that represents the server. For more information about the PEM format, go to https://en.wikipedia.org/wiki/Privacy-Enhanced_Mail.

   Ensure that the certificate file is downloaded from the mainframe in the correct format. The example in the *IBM AD Connect for Mainframe Configuration Guide* creates a DER encoded X.509 certificate that is in the Base64 format, so download this certificate in the ASCII mode if using FTP. An alternative to FTP is to cut and paste the certificate contents from the mainframe editor screen (ISPF) to a PC-based text editor, and then save the resulting file.

   Important: The personal certificate that is presented to IBM AD Build on behalf of the IBM AD Connect for Mainframe started task must have the Subject Name CN field equal to the numeric IP address of the z/OS mainframe instance where the task is running, for example,

>CN=9.30.123.456.OU=IBM CA.C=US< instead of >CN=zos.server.name.com.OU=IBM CA.C=US<. Confirm this with the person who performs AT-TLS configurations on z/OS.

3. Create a *ConnectionName*.ini file, where *ConnectionName* is the name of the z/OS connection that is defined in the **zOS Connection Setup** dialog in IBM AD Build Configuration.

   Save the file in the IBM AD Build Client bin\release folder. The full path is similar to <IBM AD Build Client installation folder>\Bin\Release. You can find a sample *ConnectionName*.ini file in the <IBM AD Build Client installation folder>\Bin\Release\Samples\AdConnectEncryptConnection\ folder.

   This .ini file follows the common INI file syntax. To enable TLS connection between IBM AD Build and IBM AD Connect for Mainframe, the **CAfile** parameter must be specified in the .ini file.

   CAfile=*CA_CERT*.pem
   Specifies the list of trusted certificates. *CA_CERT*.pem is the file name of the CA certificate or the self-signed certificate that is obtained in the previous step.

   An SSL or TLS connection uses certificates to verify the identity of the peer. During the SSL handshake, the server presents its personal certificate to the client. This certificate can be either self-signed or signed by a trusted third party such as a CA. For the SSL or TLS authentication to succeed, either the personal certificate or the CA signer certificate must be present on the trusted certificates list on the client side.

   Example

```
[OpenSSL]
#This section configures SSL/TLS connection parameters.
#It is using OpenSSL library for the communication and
#parameter syntax, wording and default values closely match those
#for OpenSSL s_client tool:
#https://www.openssl.org/docs/man1.1.1/man1/openssl-s_client.html

#A file containing trusted certificate to use for validating peer
CAfile=CA_CERT.pem
```

After setting the value of the **CAfile** parameter, the .ini file contains enough information to allow IBM AD Build Client to connect with IBM AD Connect for Mainframe by using the TLS protocol. The following list shows optional TLS-specific parameters that can be defined in the INI file. All the following parameters and the **CAfile** parameter must go under the OpenSSL section. The parameter syntax, wording, and default values closely match those for the s_client OpenSSL tool.

CAPath=C:\certs
Specifies a directory that contains multiple certificates in the hash format. More complex certificate chains can be defined in this directory.

verify=*depth*
Specifies the maximum length of a server certification chain.

cert=*client_cert*.pem
Specifies the certificate that is sent from the client to the server. Some configurations might require a client to send its certificate to the server.

certform=pem|der
Specifies the format of the certificate that is sent from the client to the server. The format can be either base64-encoded PEM or binary DER.

cert_chain=*filename*
Specifies a file that contains a set of trusted certificates for building a certificate chain. The certificate chain must start with the certificate that is specified by the **cert** parameter. A client can provide not only a single certificate but a certificate chain.

key=*private_key*.pem
keyform=pem|der
Specify the private key to the client certificate and the format of the key, which can be either PEM or DER.

pass=pass:*password*|file:*password*.txt
Specifies the password to use the private key. Private keys are usually password-protected. For an application to use the private key, you must provide a password. The password can be provided either in plain text or in a separate file. Either this INI file or the separate file must be protected by OS permissions to prevent unauthorized access.

servername=*example*.com
Specifies the expected server name. If the server is using the TLS SNI extension, the client can send the expected server name to the server. For more information about the TLS SNI extension, go to https://en.wikipedia.org/wiki/Server_Name_Indication.

tls13=true
tls12=true
tls11=true
tls1=true
Specify the maximum level of the SSL or TLS protocol that the client is willing to use. The final decision on the protocol level will be determined by the server.

bugs=true
Enables several bug workarounds for some known bugs in SSL or TLS implementations.

cipher=TLS_AES_256_GCM_SHA384
ciphersuites=TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
Specify ciphers and cipher suites to be used during communication. Custom ciphers must be supported by both the client and the server. For more information about the options and ciphers list syntax, go to https://www.openssl.org/docs/man1.1.1/man1/ciphers.html.

For more information about the s_client OpenSSL tool and the supported parameters, go to https://www.openssl.org/docs/man1.1.1/man1/openssl-s_client.html. For more information about INI files, go to https://en.wikipedia.org/wiki/INI_file.

4. Restart both IBM AD Build Client and IBM AD Build Configuration. Both applications will pick up the connection configuration changes.
5. Test the connection by clicking the **Test Connection** button in the **zOS Connection Setup** dialog in IBM AD Build Configuration.

Enabling the z/OS User Authentication feature for IBM AD Build Client

Starting with **IBM AD V6.0.1** you can use your z/OS user ID and password credentials and based on your access privileges you can pull the source files from the z/OS source systems.

Before you perform any action, you must take into account the following observations:

- Make sure that the AT-TLS setup for IBM AD Connect for Mainframe is **completed** and the TLS connection between **IBM AD Build Client** and **IBM AD Connect for Mainframe** is **enabled**. For more information, see “Enabling TLS Connection to IBM AD Build” on page 183 and “Enabling TLS Connection between IBM AD Build Client and IBM AD Connect for Mainframe” on page 291.
- Make sure that you are using the latest version of **IBM AD Connect for Mainframe**. There is no compatibility with an earlier version of **IBM AD Connect for Mainframe**.
- The **z/OS User Authentication** feature is available only for the z/OS mainframe instances that are configured with TLS and enabled in the `z0S-Data.ini` configuration file. For more information, see “Configurations” on page 295.
- All operations are performed **only** in the **graphical user interface** (**GUI**) of **IBM AD Build Client**.
- The `Please enter the z0S User/Password in order to acces the z0S Access Point` message is prompted when performing mainframe-related operations in **IBM AD Build Client**.

z/OS Username and Password Guidelines

- For the **IBM AD V6.0.1** release, only a standard of 8 character mixed case password is supported. Password phrases are not supported.
- The username and password are volatile and they are kept in the memory for the running instances of **IBM AD Build Client** and **IBM AD Build Configuration**.
- If you close **IBM AD Build Client** and later you want to perform any mainframe-related operation it will be necessary to reenter the username and password. Closing the application is considered a reset process for the z/OS credentials. It shall be used in case of wrong or modification of the z/OS credentials.
- In case that a project has multiple z/OS mainframe instances available, the username and password is requested for each of the z/OS mainframe instance that is used in the related-mainframe operation and only once per **IBM AD Build Client** instance.

  Note: If other projects have the same z/OS mainframe instance attached and the z/OS mainframe instance was already prompted in a project, it will not be subsequently prompted for other projects that are on the same z/OS mainframe instance.
- If a mainframe-related operation is started on a selection of multiple z/OS mainframe instances and if you cancel entering the username and password for one of the z/OS mainframe instances, the whole operation will be aborted.

Configurations

To enable z/OS User Authentication feature, perform the following configurations:

1. Go to `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\Samples` and copy the `zOS-Data.ini` configuration file in the `<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release` folder.
2. Open the `zOS-Data.ini` configuration file by using a text editor and enter the desired values for the parameters that are detailed below. By default, the `zOS-Data.ini` configuration file has the following format:

   ```
   [MVS1]
   use_cred=Y/N

   [MVS2]
   use_cred=Y/N
   ```

   Where:

   - **[MVS1]** and **[MVS2]** are the names of the z/OS mainframe instances that are present in **IBM Application Discovery Build Configuration** > **zOS** Tab.

     Important: The square brackets need to be preserved and between them, the actual name of the z/OS mainframe instance needs to be added with no spaces nor tabs between the brackets.
   - **use_cred** is an attribute that expects the **Y** or **N** values. This attribute decides whether the mentioned z/OS mainframe instance is prompted for the username and password or not.

     Note: The attribute value is not case-sensitive.

   Important: In case that a z/OS mainframe instance is not present in the configuration file, or it is present but the attribute value is set to **N**, for the related z/OS mainframe instance, the `Please enter the zOS User/Password in order to acces the zOS Access Point` message will **not** be prompted when performing mainframe-related operations in **IBM AD Build Client**.
3. Save and close the `zOS-Data.ini` configuration file.

Example of a `zOS-Data.ini` configuration file

```
[zOS_A]
use_cred=Y
```

```
[zOS_C]
use_cred=y

[zOS_D]
use_cred=N
```

Where:

- **zOS_A**, **zOS_B**, **zOS_C** and **zOS_D** are the names of the z/OS mainframe instances.
- The feature is available only for **zOS_A** and **zOS_C**.

Using the z/OS User Authentication feature

Once the z/OS mainframe instance is configured with TLS and declared in the `zOS-Data.ini` configuration file, in **IBM AD Build Client** you can perform the following operations:

1. **Add Files from Mainframe**
   a. In **IBM® Application Discovery Build Client**, open an already existing project, right-click on a virtual folder, and select **Add Files from Mainframe**.
   b. The `Please enter the zOS User/Password in order to acces the zOS Access Point` message is prompted when performing the **Add Files from Mainframe** operation.
   c. Enter the username and password for the related z/OS mainframe instance. For more information, see “z/OS Username and Password Guidelines” on page 295.
   d. Perform all the necessary actions related to the **Add Files from Mainframe** operation. For more information, see “Adding Files From Mainframe Library” on page 263.
2. **Update Modified Mainframe Members**

   Note: The Synchronize Mainframe Members feature must **not** be activated to have the **Update Modified Mainframe Members** option available. For more information, see “Synchronizing Mainframe Members” on page 266.

   a. In **IBM® Application Discovery Build Client**, open an already existing project, right-click on a virtual folder, and select **Update Modified Mainframe Members**.
   b. The `Please enter the zOS User/Password in order to acces the zOS Access Point` message is prompted when performing the **Update Modified Mainframe Members** operation.
   c. Enter the username and password for the related z/OS mainframe instance. For more information, see “z/OS Username and Password Guidelines” on page 295.
   d. Perform all the necessary actions related to the **Update Modified Mainframe Members** operation. For more information, see “Updating Projects” on page 266.
3. **Synchronize members**

   Note: The Synchronize Mainframe Members feature **must be activated** to have the **Synchronize members** option available. For more information, see “Synchronizing Mainframe Members” on page 266.

   a. In **IBM® Application Discovery Build Client**, open an already existing project, right-click on the project, and select **Synchronize members**.
   b. The `Please enter the zOS User/Password in order to acces the zOS Access Point` message is prompted when performing the **Synchronize members** operation.
   c. Enter the username and password for the related z/OS mainframe instance. For more information, see “z/OS Username and Password Guidelines” on page 295.
   d. Perform all the necessary actions related to the **Synchronize members** operation. For more information, see “Synchronizing Mainframe Members” on page 266.

IBM AD Build CLI Commands

I. Overview

In the following sections, a list of **IBM AD Build Client** and **IBM AD Build Configuration** batch commands is presented. These commands are used to cover a set of actions such as **Create Projects**, **Upgrade Repositories Structures**, **Build Projects**, **Synchronize Projects**, using CLI commands.

The list of actions that can be performed in the background, use cases and best practices are also available in HTML format.

1. Click **Start**, select **Run** then type **cmd** to open the command window.
2. Go to the folder where your **IBM AD Build Client** is installed and locate `IBMApplicationDiscoveryBuildClient.exe`.
3. Drag `IBMApplicationDiscoveryBuildClient.exe` into the command window then type **/?** and press **ENTER**.

As a result, a web page is displayed containing detailed information.

II. Description of the IBM AD Build Client Batch Commands

1. Creating a new project in background

IBM AD Build Client can be invoked by using two parameters as follows:

```
IBMApplicationDiscoveryBuildClient.exe /np Full path to the .ini configuration file
```

Where:

- **/np** is the parameter responsible for creating a new project.
- The **.ini configuration file** is the file in which the details for creating a new project are defined. The full path to the `.ini` file must be used as a parameter.

The content of the `.ini` file is defined as follows:

Note: The values of the parameters that are present in the **.ini configuration file**, need to be **identical** to the ones present in the **New Project** window, from **IBM AD Build Client**. The only exception is for the **ProjectDBType** parameter, where the value needs to be **SQL**, **DB2** or **DB2_LUW**.

```
[ADNewProj]
ProjectName = "name of the project"
Path = "full path of the project including the project's name"
Environment = "zOS"
ProjectLanguages = "DT Cobol,Assembler,Cobol,Natural,PL1,Ads"
DBTypes = "Datacom,IDMS,Adabas,Relational,IMS/DB"
MapTypes = "Natural (LNM),CICS (BMS), IMS TM (MFS),ADS Map"
ProjectDBType = "SQL" or "DB2" or "DB2_LUW"
CCSEnvironment = "Environment name"
DBServerName = "DB Server Name blank space [ip/name:port]"
AttachToDB = "Y" or "N"
DBName = "dbName"
SchemaName = "schemaName"
EnableBRD = "Y" or "N"
EnableCross = "Y" or "N"
```

Notes®:

1. **[ADNewProj]** is the section name and under any circumstances it must **not** be modified.
2. The **Lines** from the `.ini` file can be commented by adding "**;**" (semicolon) at the beginning of each line.
3. The **names** of the following parameters must **not** be modified:
   - ProjectName

- Path
- Environment
- ProjectLanguages
- DBTypes
- MapTypes
- ProjectDBType
- CCSEnvironment
- DBServerName
- AttachToDB
- DBName
- SchemaName
- EnableBRD
- EnableCross

4. The **values** for all the parameters must be added between double quotation marks.
5. The **Path** parameter must contain **as value** the full path of the project (the project name included in the path).
6. The **CCSEnvironment** parameter takes as value the **Environment name** as defined in **IBM AD Configuration Server** at the following location: **Home Page** > **Configuration server name** > **Environments** > **"MyEnvironmentName"**.
7. The **DBServerName** parameter takes as value the **Relational database server name** and the **host name or the IP of the computer, including the port** as defined in **IBM AD Configuration Server**. The value for the **DBServerName** parameter is case-sensitive and has the following format:

   `"Relational Database Server Name` *`a mandatory blank space`* `[ip/hostname:port]"`

   Where:

   - **Relational Database Server Name** is defined in **IBM AD Configuration Server** at the following location: **Home Page** > **Configuration server name** > **Environments** > **"MyEnvironment"** > **Relational Database Servers**.
   - **[ip/hostname:port]** is defined in **IBM AD Configuration Server** at the following location: **Home Page** > **Configuration server name** > **Environments** > **"MyEnvironment"** > **Relational Database Servers** > **"MyRelationalDBServer"**. The information is present under the **Host** and **Port** entries.

   Example:

   ```
   DBServerName = "SQLPROD [SQL.PROD.HOST:1433]"
   ```

8. The **AttachToDB** parameter can be set to *Y* or *N* as follows:
   - **Must be set to N** when you create a project on:
     - SQL Server.
     - Db2 on z/OS or Db2 for LUW, where the project's database doesn't exist and needs to be created.
   - **Must be set to Y** when you use **Db2 on z/OS or Db2 for LUW as a relational database server** and the project you are about to create has a Database and a Schema already created.
9. The **DBName** and **SchemaName** parameters can be configured as follows:
   - When using **Db2 on z/OS as a relational database server** and **AttachToDB** is set to **Y**, you must enter the corresponding values for the **DBName** and **SchemaName** parameters.
   - When using **Db2 for LUW as a relational database server**, it is mandatory to enter the corresponding value for the **DBName** parameter. The **SchemaName** parameter's value is hardcoded to **EZSCH** and it cannot be changed.

Note: When using **SQL Server** as a relational database server, the **AttachToDB**, **DBName**, and **SchemaName** parameters do not expect any value.

10. The **EnableBRD** parameter can be set to *Y* or *N* as follows:
    - **Must be set to N** when the project is not part of the BRD analysis.
    - **Must be set to Y** when the project is part of the BRD analysis.
11. The **EnableCross** parameter can be set to *Y* or *N* as follows:
    - **Must be set to N** when the project is not part of the cross project analysis
    - **Must be set to Y** when the project is part of the cross project analysis.

      Note: This option is not yet available for Db2 on z/OS project database type. Setting this on Y for Db2 on z/OS projects will result in error.

For a better understanding and examples, see the Use Cases and Best Practices section.

A sample for the **New Project** `.ini` file is available at the following location: `<IBM AD Build Client installation folder>\Bin\Release\Samples\Background_Actions\newProjectBASample.ini`.

The logs for **New project in background** are available at the following location: `<IBM AD Build Client installation folder>\Bin\Release\NewProjInBackLog_timestamp.log`.

2. Repository upgrade

IBM AD Build Client can be invoked in batch mode to upgrade the project's repository (the database's structure) using the following command:

```
IBMApplicationDiscoveryBuildClient.exe /ru  LOG file name  optional parameter: full path to a
file that contains the list of the projects
```

Where:

- **/ru** is the parameter responsible for upgrading the project's repositories (Upgrade Repository).
- **LOG File Name** is the full path of the log file responsible to show the status and progress for the **Repository Upgrade** process. This log file can have any name and may be in any given path (as long as you have read/write access to that path).
- **Optional Parameter: Full path to a file that contains the list of the projects to be upgraded** is the parameter that must be used only when a given number of projects must be upgraded (not all projects are upgraded). This file must contain the name of the projects (as defined in **IBM AD Build Client**) that get their repositories upgraded (one project per line).

  Example for how the optional parameter file looks like:

  ```
  Project1
  Project2
  Project3
  ```

  Note: If the optional parameter is not set, then all of the existing projects will have their repositories upgraded.

For a better understanding and examples, see the Use Cases and Best Practices section.

A sample for the **Repository Upgrade** optional parameter file (list of projects) is available at the following location: `<IBM AD Build Client installation folder>\Bin\Release\Samples\Background_Actions\repositoryUpgradeBASample.txt`.

The logs for **Repository Upgrade** are available in the location that is mentioned in the description of **LOG File Name**.

3. Build project in background

IBM AD Build Client can be invoked in batch mode to run a build (full build) by using the following command:

```
IBMApplicationDiscoveryBuildClient.exe /fb ProjectName
```

Where:

- **/fb** is the parameter that is used to invoke the build action.
- **ProjectName** is the name of the project where full build is triggered.

For a better understanding and examples, see the Use Cases and Best Practices section.

The log for **Build in background** is similar to Build in GUI mode. The log can be found under the project's folder as project's name and timestamp. Example: `MyProject1_timestamp.txt`.

4. Periodic Updates for IBM AD Build Projects

IBM Application Discovery projects must be kept up-to-date in terms of the source code and the information that is stored in the repository. For such a use case, **IBM AD Build Client** can be invoked in batch mode by using two different commands as follows:

4.1 To keep the source code up-to-date in the projects by using the **Synchronization** process.

```
IBMApplicationDiscoveryBuildClient /umm1 ProjectName
```

Where:

- **/umm1** is the parameter that is used to invoke the Synchronization process.
- **ProjectName** is the name of the project where the Synchronization process is triggered.

Note: It is mandatory to have the **Synchronize** feature *enabled* in **IBM AD Configuration Server** and the synchronization file configured. To *enable* the **Synchronize** feature, go to **Home Page** > **Install Configurations** > **IBM Application Discovery Build Client** > **Enable Members Synchronization**.

For a better understanding and examples, see the Use Cases and Best Practices section.

A sample for the **Synchronization** file is available at the following location: `<IBM AD Build Client installation folder>\Bin\Release\Samples\Background_Actions\synchronizeBASample.txt`.

The logs for **Synchronize** are available under: **Project's Folder** > **Synchronize**.

4.2 To keep the information stored in repository up-to-date by using the **Make** process.

```
IBMApplicationDiscoveryBuildClient /m1 ProjectName /m2 y /m3 y
```

Where:

- **/m1** is the parameter that is used to invoke the Make process.
- **ProjectName** is the name of the project where the Make process is triggered.
- **/m2 (y/n)** refers to whether the make process is forced or not as follows:
  - **/m2 y** means that if another **AD Component is using the project in read mode**, the process starts.
  - **/m2 n** means that if another **AD Component is using the project in read mode**, the process does not start until the project is released.
- **/m3 (y/n)** refers to whether the status of the Make process is logged or not as follows:
  - **/m3 y** means that the **status log file `BatchMakeStatusFile_timestamp.txt`** is generated under the **project's folder**.
  - **/m3 n** means that the **status log file** is not generated.

For a better understanding and examples, see the Use Cases and Best Practices section.

The logs for the **Make** process in background can be found under the project's folder as `BatchMakeStatusFile_timestamp.txt` and `Project'sName_timestamp.text`.

5. Automatically check the projects for missing components in IBM AD Build Client

After adding source files to a project, you might want to know if the project is complete in terms of dependencies. IBM AD Build Client can be invoked in batch mode to check the projects for missing components (similar to the 'Check' option that can be found in GUI Mode) by using the following command:

```
IBMApplicationDiscoveryBuildClient.exe /cp ProjectName [A,B,C,D,1,2,3,4,5,6]
```

Where:

- **/cp** is the parameter that is used to invoke the **Check project** in batch mode.
- **ProjectName** is the name of the project to be checked.
- **[A,B,C,D,1,2,3,4,5,6]** defines the scope of the check, as follows:

  Important: If the brackets are missing, **all the components beneath** are checked.

  - **Letters specify the referenced resources:**
    - A - Programs
    - B - Maps
    - C - Programs called in jobs
    - D - ADS Dialogs
  - **Digits specify the critical components:**
    - 1 - Procs
    - 2 - Program Include Files
    - 3 - JCL Include Files
    - 4 - Control Files
    - 5 - Transactions
    - 6 - ADS Processes

Important: Make sure that there are no spaces between the scope elements in the command line. Each space is considered a CLI parameter delimiter. A space can break the set and it can result in having an invalid command line.

6. Update API Resolutions usage

IBM AD Build Client can be invoked in batch mode to update **API Resolutions** by using the following command:

```
IBMApplicationDiscoveryBuildClient /uar1 ProjectName
```

Where:

- **/uar1** is the parameter that is used to invoke the update of API Resolutions.
- **ProjectName** is the name of the project where the API Resolutions process is triggered.

Note:

- It is mandatory to have the **API Extensibility** feature *enabled* in **IBM AD Build Client** . To *enable* the **API Extensibility** feature, go to **IBM AD Build Client** > **Project** > **Settings** > **Extensibility** > **Enable API/Macro handling by using a configuration file**.
- The **Update API Resolutions** must be used for the situation when the API Resolution File is modified, so the resolutions must be updated in the repository but without running a full build on the project (which might be time consuming).

The logs for the **Update API Resolutions** usage are available under: **Project's Folder** > **UpdateApiResolutions_timestamp.txt**.

III. Description of the IBM AD Build Configuration Batch Commands

IBM AD Build Configuration can be invoked in batch mode by using the following command:

```
IBMApplicationDiscoveryBuildConfiguration.exe /ba Full path to the corresponding .ini
configuration file
```

Where:

- **/ba** is the parameter that is used to invoke it in batch mode.
- The **.ini configuration file** is the file in which the actions that are executed in batch mode are defined. The full path to the `.ini` file must be used as a parameter.

The content of the `.ini` file is defined as follows:

```
[ProjectsActions]
DeleteProject=PjName1,PjName2
AssociatezOS=[PjName1, zOSConnName1],[PjName2, zOSConnName2]
[OperationalInformation]
CA7IMPORT=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
CA7UPDATE=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
TWS=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
CSD=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
IMS=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
Adabas=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
DB2=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
Predict=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
SMF=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
MQ=[PjName1, zOSConnName1],[PjName2, zOSConnName2]...
```

Notes:

1. **[ProjectsActions]** and **[OperationalInformation]** are the section names and under any circumstances they must **not** be modified.
2. The **Lines** from the `.ini` file can be commented by adding "**;**" (semicolon) at the beginning of each line.
3. The **names** of the following parameters must **not** be modified:
   - DeleteProject
   - AssociatezOS
   - CA7IMPORT
   - CA7UPDATE
   - TWS
   - CSD
   - IMS
   - Adabas
   - DB2
   - Predict
   - SMF
   - MQ
4. All the parameters that are mentioned in step 3 accept the following values:
   - **PJName()** is the name of the project.
   - **zOSConnName()** is the z/OS Connection's name that is associated to the project in **IBM AD Build Configuration**.

5. For the parameters that accept two values, such as Associate z/OS and all Operational Information, it is mandatory to keep "**[ ]**" (square brackets) as described in the above template.

For a better understanding and examples, see the Use Cases and Best Practices section.

A sample for the `.ini` file is available at the following location: `<IBM AD Build Client installation folder>\Bin\Release\Samples\Background_Actions\BuildConfigurationBASample.ini`.

The logs for **IBM AD Build Configuration** batch commands are available at the following location: `<IBM AD Build Client installation folder>\Bin\Release\Log\ADBuildConfiguration_timestamp.log`.

IV. Use Cases and Best Practices

Make sure that **IBM Application Discovery** is **installed, configured, and up and running** before you get into these use cases.

All the **configurations (files)** that are considered as prerequisites to run the batch commands are put in place and below you can find some best practices and samples that can be applied in any environment.

Details on how and when to use these batch commands are provided and **you can use any scripting method to invoke and execute**, entirely based on your needs.

Note: It is important to follow the same order as shown below when you invoke the **IBM Application Discovery** batch commands.

1. Scheduling periodic updates for IBM AD Build Client projects

a. IBM ADDI - GIT server integration

Prerequisites:

- **Any Git Client** must be installed on the machine where IBM AD Build Client is located.
- The GIT Repo(s) that contain the source code and is about to be loaded/analyzed in ADDI, must be cloned and available for IBM AD Build Client.

Create a batch file that triggers the following actions:

- **Synchronize** (update, add, delete) the existing members in projects **MyProject1**, **MyProject2**, and **MyProject3** so that they are up-to-date.
- Run the **Make** process so that only the modified members (changed, added) are built and the information is refreshed in the repository.

The content of the `.bat` file:

```
REM  ------------------------  Get the latest sources from Git -----------
cd <Location of the Git Repository>
git pull

REM ------------------------ Run the local SYNC process ----------------------
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject1
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject2
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject3

REM ----------------------------- Run Make---------------------------------------
IBMApplicationDiscoveryBuildClient.exe /m1 MyProject1 /m2 y /m3 y
IBMApplicationDiscoveryBuildClient.exe /m1 MyProject2 /m2 y /m3 y
IBMApplicationDiscoveryBuildClient.exe /m1 MyProject3 /m2 y /m3 y
```

Note:

- **For the example above**, it is assumed that all 3 projects are part of the same GIT Repository, therefore a single ***GIT Clone*** command is executed.

- **The Synchronization File** please make sure it is configured to execute Local Synchronization, as it is mentioned in the distributed sample here: ***<IBM AD Build Client installation folder>\Bin\Release\Samples\Background_Actions\synchronizeBASample.txt.***
- **The `.bat` file can be scheduled** by using, for example, Windows Scheduler, so that it can be triggered on a daily, weekly, or monthly basis (depending on how often the source is changed and needs to be updated). For more information, see "V. Setting up Automatic Updates with Windows Scheduler" on page 305.
- **The Synchronize and Make processes**, depending on the number of changes that are processed, they may be time consuming, therefore, based on your environment, set the scheduler to run when the users are not using the projects for analysis.

b. IBM ADDI - Integration with Source Control Management Systems based on the host(CA Endevor, Changeman ZMF) and with PDS Libraries

Create a batch file that triggers the following actions:

- **Synchronize** (update, add, delete) the existing members in projects **MyProject1**, **MyProject2**, and **MyProject3** so that they are up-to-date.
- Run the **Make** process so that only the modified members (changed, added) are built and the information is refreshed in the repository.

The content of the `.bat` file:

```
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject1
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject2
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject3
IBMApplicationDiscoveryBuildClient.exe /m1 MyProject1 /m2 y /m3 y
IBMApplicationDiscoveryBuildClient.exe /m1 MyProject2 /m2 y /m3 y
IBMApplicationDiscoveryBuildClient.exe /m1 MyProject3 /m2 y /m3 y
```

Note:

- **The `.bat` file can be scheduled** by using, for example, Windows Scheduler, so that it can be triggered on a daily, weekly, or monthly basis (depending on how often the source is changed and needs to be updated). For more information, see "V. Setting up Automatic Updates with Windows Scheduler" on page 305.
- **The Synchronize and Make processes**, depending on the number of changes that are processed, they may be time consuming, therefore, based on your environment, set the scheduler to run when the users are not using the projects for analysis.

2. Automatically upgrade projects repositories and run a build as soon as a new version of IBM AD Build Client is installed

Create a batch file that triggers the following actions:

- **Upgrade the repository** for projects **MyProject1**, **MyProject2**, and **MyProject3** (found in the `RepListofProjects.txt` configuration file).
- **Execute a build (full build)** on the mentioned projects.

The content of the `.bat` file:

```
IBMApplicationDiscoveryBuildClient.exe /ru  "\\AppServer\IBMAD\Conf\Logs\RepUpgrade.log" "\
\AppServer\IBMAD\Conf\RepListofProjects.txt"
IBMApplicationDiscoveryBuildClient.exe /fb MyProject1
IBMApplicationDiscoveryBuildClient.exe /fb MyProject2
IBMApplicationDiscoveryBuildClient.exe /fb MyProject3
```

3. End to end flow in IBM AD Build: Create new projects, associate an existing z/OS Connection to projects, get the source code based on the host in PDS libraries and Endevor SCM (by using the Synchronize feature), build projects

Create a batch file that triggers the following actions:

- **Create three new projects**: **MyProject4**, **MyProject5**, and **MyProject6** (as configured in `MyProject4.ini`, `MyProject5.ini` and `MyProject6.ini`).
- **Associate the existing z/OS Connection**: **Lpar1** to the newly created projects (as defined in `zOSConf.ini`).
- **Get the source code from the host** for the mentioned projects by using the **Synchronization** feature (as configured in `SyncProjectConf.txt`).
- **Run a build (full build)** on the mentioned projects.

The content of the `.bat` file:

```
IBMApplicationDiscoveryBuildClient.exe /np "\\AppServer\IBMAD\Conf\MyProject4.ini"
IBMApplicationDiscoveryBuildClient.exe /np "\\AppServer\IBMAD\Conf\MyProject5.ini"
IBMApplicationDiscoveryBuildClient.exe /np "\\AppServer\IBMAD\Conf\MyProject6.ini"
IBMApplicationDiscoveryBuildConfiguration.exe /ba "\\AppServer\IBMAD\Conf\zOSConf.ini"
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject4
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject5
IBMApplicationDiscoveryBuildClient.exe /umm1 MyProject6
IBMApplicationDiscoveryBuildClient.exe /fb MyProject4
IBMApplicationDiscoveryBuildClient.exe /fb MyProject5
IBMApplicationDiscoveryBuildClient.exe /fb MyProject6
```

4. Delete one or a set of projects in batch mode

Run as standalone or create a batch file that deletes projects **MyProject4** and **MyProject6** (as configured in `DeletePj.ini`).

```
IBMApplicationDiscoveryBuildConfiguration.exe /ba "\\AppServer\IBMAD\Conf\DeletePj.ini"
```

The content of the `DeletePj.ini` file:

```
[ProjectsActions]
DeleteProject=MyProject4,MyProject6
```

V. Setting up Automatic Updates with Windows Scheduler

The Windows Scheduler can be used to run automatic, periodic updates to make sure that the resources you are working on are always up-to-date.

1. To set up the automatic updates in **Windows Scheduler**, select **Start** > **Control Panel** > **Administrative Tools** > **Task Scheduler**. If you're prompted for an administrator password or confirmation, type the password or provide the confirmation.
2. Select the **Action** menu then click the **Create Basic Task**. Type a name for the task and an optional description then click **Next**.
3. Do one of the following actions.
   - To select a schedule based on the calendar, click **Daily**, **Weekly**, **Monthly**, or **One time**, click **Next**; specify the schedule that you want to use, and then click **Next**.
   - To select a schedule based on common recurring events, click **When the computer starts** or **When I log on**, and then click **Next**.
   - To select a schedule based on specific events, click **When a specific event is logged**, then click **Next**. Specify the event log and other information by using the menu lists, and then click **Next**.
4. To schedule a program to start automatically, click **Start a program**, and then click **Next**.
5. Click **Browse** to find the program you want to start, and then click **Next**. Click **Finish**.

   Note: The `*.bat` file present in the "IV. Use Cases and Best Practices" on page 303 section can be used in the scheduler.

Chapter 8. Analyzing Applications with IBM AD Analyze

The objective of this section is to provide the information needed to use IBM AD Analyze and to understand its capabilities.

Concepts and capabilities of IBM AD Analyze are presented, including a description of the steps and procedures typically followed in IBM AD Analyze Client to set up and analyze a system and to view the results. This section also contains a detailed description of the interface as well as a presentation of the Server Manager application.

Projects, Folders, and Files

Use **Projects, Folders, and Files** to organize projects, folders, and files in IBM AD Analyze Client.

IBM AD Analyze Client projects correspond to independent applications. An IBM AD Analyze Client project can contain references to all application source files or part of them. These are organized into folders that are category lists for the different kinds of files that make up the project/application. Standard folders are defined and included in the project by default. Depending on the environment of the application to be analyzed, projects can have different folder structures. New folders can be added if necessary. Creating an IBM AD Analyze Client project creates a project folder on the computer or on a network drive. You can specify the location for this folder.

IBM AD Analyze Client tasks

Working with **IBM AD Analyze Client** usually means performing the following sequence of tasks:

Table 8. IBM AD Analyze Client tasks

Start IBM AD Analyze Client	
Create a New Project	Create a project using the IBM AD Analyze Client wizard by specifying the project name, location, and folders.
Adjust Project Settings	Adjust the project settings as appropriate.
(Re)build	Build the project.
Analyze	Examine the results of a build by means of graphs, reports, usage and compare analyses.

Tasks 2 and 3 are setup and organizational steps. Their purpose is to define the source material that is analyzed. Step 4 (Build) populates the repository that is the basis for the Analysis step.

UI Reference

Perspectives

© Copyright IBM Corp. 2010, 2022

Application Discovery Analysis Perspective

Application Discovery Analysis and **Application Discovery Flow** perspectives contain the same tabs. The only difference is the way in which the tabs are organized on your screen.

In **Application Discovery Analysis** perspective, the Project Explorer or **AD Navigator tab** displays a hierarchical tree that is comprised of all the projects and project components that are loaded into **IBM AD Analyze Client**. The tree nodes can be expanded or collapsed by clicking the "+" or "-" signs to the left of each node. The type of each branch is identified by an icon and a text label. Each node corresponds to a source code file or XML meta-data file of a project entity. Double-clicking the node causes the source file or XML to be displayed in the editor window. Right-clicking a node causes a pop-up menu to open. The pop-up menu usually contains commands for displaying the code, expanding or collapsing the branch represented by the node, or for viewing properties of the object.

AD Navigator Toolbar

Many of the AD Navigator Toolbar options are Eclipse default options. Here, are presented only the options that are relevant for **AD Analyze Client**. For details about the other options, refer to your Eclipse help files.

The following options are available, depending on the selected project/node:

Table 9. AD Navigator toolbar buttons

Function	Description
Back	Takes you back to the previous view.
Forward	Takes you to the next view.
Up to Repository	Displays the contents of the root directory.
Collapse All	Collapses all the branches of the Analysis Navigator tree.
Link Open Editors with Content in the Navigator	Toggles on/off the link between the selection in the Navigator view and the active editor.
Menu	Displays a menu of items available to the Navigator.

Project Node

In **AD Navigator** tab, double-click the Project node to expand/collapse the project tree. To display the options available for the project node, right-click on the selected node to display the corresponding pop-up menu. The options available in the project pop-up menu vary according to the type of project selected (Java, CSharp, or NetWeaver). For details on the organizational entities in a project, see Projects Folders and Files.

Following are explanations of menu options available in the project node pop-up menu for Java projects. For more information on on the other options, see the Eclipse Help.

Table 10. Java Project node pop-up menu

Menu item	Explanation
Call Graph	Shows the call graph that presents the chain of possible calls at runtime in Java.
Resource Dependencies Graph	Presents the Resource Dependencies graph showing the links between the special entities present in the special web category (objects such as *.jsp, *.js, *.css, *.html files). For details, see Resource Dependencies Graph in the Graphs chapter.

Table 10. Java Project node pop-up menu (continued)	
Menu item	**Explanation**
Entity Diagram	Presents a diagram for all the classes in the project that contain an “@Entity” annotation. For details, see Entity Diagram in Graphs chapter.
Imports Analysis Graph	Presents the inheritance relationships (imported, inherited, implemented) among all the entities in the application analyzed. For details, see Imports Analysis Graph in the Graphs chapter.
Third-Party Usage Report	Presents a list of the classes that use third-party entities (classes and packages).
Java Complexity Report	Presents complexity measurements that are developed to render the complexity of a class directly from source code.
McCabe Complexity Report	Presents the **McCabe Complexity Report** showing McCabe complexity for the current project or for the selected object.
Java Inventory Report	Presents a list of all the components in the current project.
Cross Applications Call Graph	Presents relationships that exist between Java and Cobol applications.

Project Properties

To access the project properties, right-click a project in the **AD Navigator** tab then from the pop-up menu that appears select Properties. A dialog window is displayed offering the following options: **Project folders**, **Project database settings**, and **External classpath**.

Note: The available options depend on the type of project selected.

You can use the available tabs to set the type of project folders, set up or change the project database configuration or determine the external classpath.

Changing the project database settings

Note: For multi-user installations, only the owner of a project can change the database and the project folders settings. For all the other users, the fields in the **Project database settings** and **Project folders** tabs are disabled.

The user can modify the database settings of a project at any time. For Multiuser installations, restrictions apply. If another analysis is in the progress, no change is allowed in the database settings. To change the project database settings, select the project from the AD Navigator, right-click to display the corresponding pop-up menu, and select **Properties**.

The Project database settings window is displayed. You can change the server host and port, you can create a new database, or you can connect to another existing database. Or you can make the database public by clearing the selection of **Restrict access for new projects** option. Use **Test Connection** to verify the connection with the new database. For detailed instructions on how to set up a new database, see Creating a Java project.

Folder Type Nodes

Each folder in the Analysis Navigator tab represents a list of source files of the corresponding type that are included in the project. This list is displayed when the node is expanded. Double-click the folder node to expand/collapse the project tree.

By default, a Java project is split into two groups of folders, which represent the two types of components of a Java application: src (Java source folder) and lib (third-party libraries folder). An extra web folder (web sources folder) is available, it stores web application resources of your Java project (if applicable).

Component Node

Double-click a component node to display the component. In case the component has Java source code, the source is opened in the Editor. In case the component has only attributes without a source code, an XML representation of the component is opened in the Editor. To better see attributes of a component, choose the Properties view and then select the component in the AD Navigator instead of opening the XML.

Application Discovery Browser Perspective

Use **Application Discovery Browser Perspective** to view and analyze mainframe projects created with **IBM AD Build Client**. Other types of projects such as Java and C language projects can be also viewed and analyzed using this perspective. From the perspectives list (**Windows** > **Open Perspective** > **Other**), select **Application Discovery Browser** to display it as in the following image.

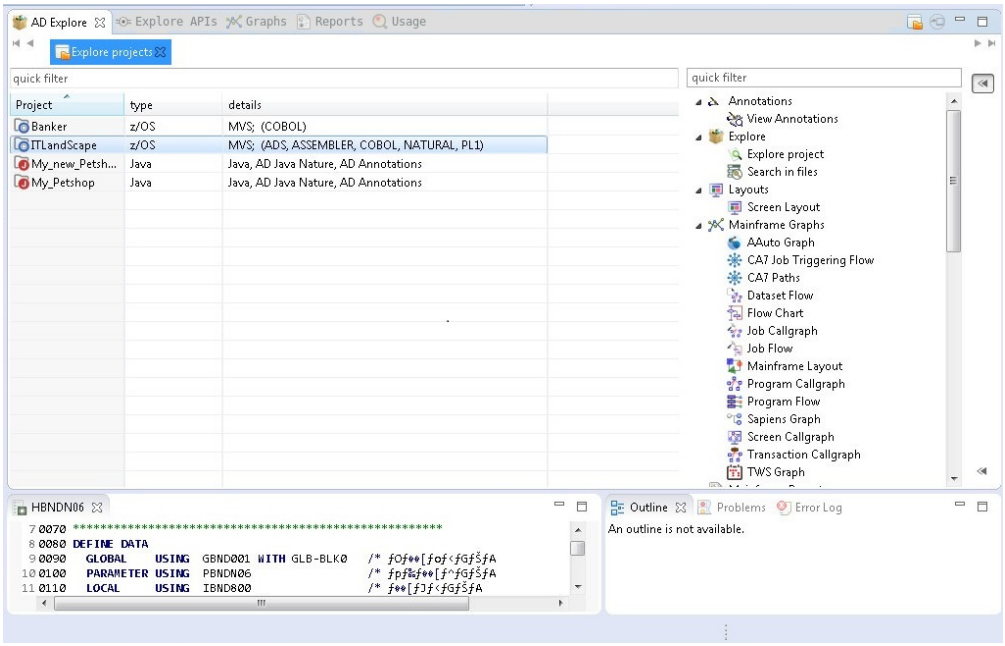


Figure 34. Application Discovery Browser Perspective

The central part of the screen - the **AD Explore/Explore Projects** tab - displays a list of projects found on the **IBM AD Build Configuration** (other types of projects will only be visible in this list if created within **Application Discovery Analysis** perspective or if imported from **IBM AD Analyze Server**).

Apart from AD Explore, the following tabs are also available: **Explore API**, **AD Graphs**, **AD Usage**, and **AD Reports**.

AD Explore Tab

Explore Projects Tab

Use **Explore projects** to analyze mainframe projects that are created with **IBM AD Build Client** as displayed as in the following image.

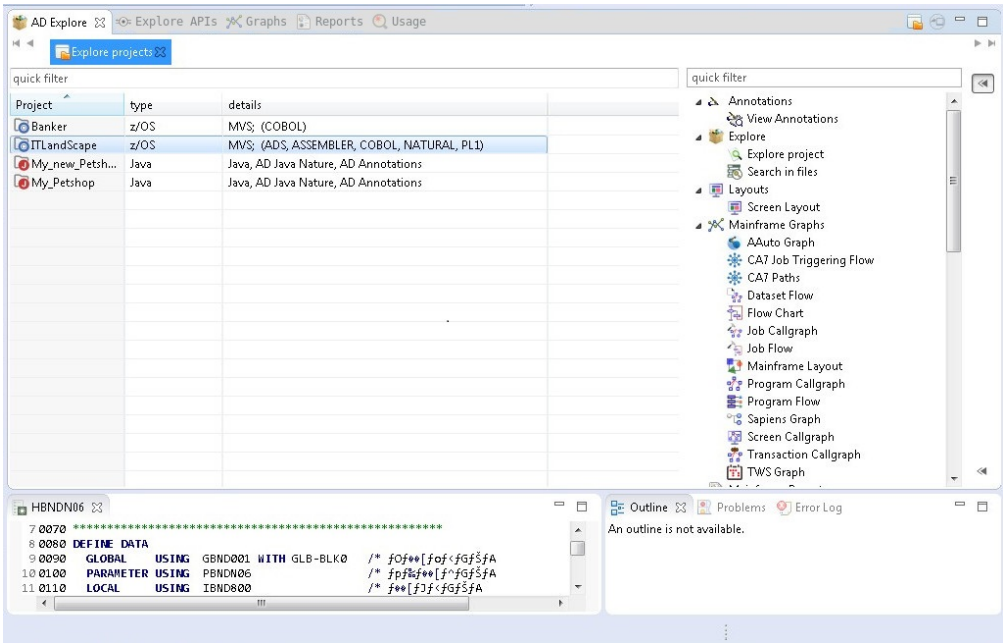


Figure 35. Explore Projects tab

In the central part of the screen, the Explore Projects view within **AD Explore** tab, displays a list of projects that are found on the IBM AD Build Configuration. For each project, the corresponding type is displayed. The Details column presents the type of environment in which the project was created and the types of resources that are included in the project. The pane to the right of the screen presents the types of graphs, reports, and usage analyses, which can be generated for the selected project.

When **IBM AD Analyze Client** starts, the list of the mainframe projects is empty. To have the list of mainframe projects available it is necessary to use the **Get project list** contextual-menu option by right-clicking in the **Explore projects view**.

If **IBM AD Mainframe Projects Service** cannot be contacted to get the list, a specific error is displayed in the **Error Log** area. Periodically, the service is asked for the updated list, being easy to identify if a mainframe project was added or removed.

If a user that uses an environment where authentication is enforced, the call for the list of projects should be made only after a successful authentication. Otherwise, the list of the mainframe projects remains empty.

Note:

- All users can have access to all projects present in an environment if authentication is not needed and if the mainframe projects are part of public workspaces.
- Starting with **IBM AD V5.1.0.4**, the list of projects is not cached at the restart of the **IBM AD Analyze Client** and it is necessary to use the **Get project list** contextual-menu option each time when **IBM AD Analyze Client** starts.

The types of graphs, reports, and usage analyses available in the pane to the right of the screen depend on the nature of the selected project and on the number of projects.

- For details on the analyses available for mainframe projects, see Mainframe Graphs, Mainframe Reports, and Usage.
- For details on the analyses available for a Java project, see Java Graphs and Java Reports.
- For details on the analyses available when two projects are selected, see Compare Projects in Editor, Compare Projects Report, Shared Resources Graph, Shared Resources Report.

In the upper right corner of **AD Explore** tab the following buttons are available:

- **Explore projects**, click this button to return to the Explore projects tab.

- **Move to separate view**, click this button to move the current view to a separate tab.

Explore API

Use this view as a starting point of analysis regarding APIs defined the z/OS Connect.

When first starting the **Analyze Client**, the view is empty; at the first call of *Reload content* action (right-click on tool-bar) the view will be filled with data read from **AD Catalog** using the web-service settings from **Preferences**. This list is cached and kept on client until the next call of the *Reload* action. By selecting an API/service, corresponding analyses are visible in the right part of the screen. Available analyses at the moment are: "API Usage" on page 424 and "API Call Graph" on page 332. Furthermore, when selecting an API/service, its details are visible in **Properties** view.

Limitations of analysis made on APIs/services.

- services using the **WOLA** provider are not supported yet.
- services that are not exposed through APIs are not visible and cannot be analyzed.

Manage Resolutions Analysis Tab
Manage Resolutions analysis tab in **Application Discovery Browser** perspective is described in detail.

Note: This function is only available for mainframe projects for which a graph database was set up. Currently, the Manage Resolutions function is implemented only for Cobol and Assembler programs.

The **Manage Resolutions Analysis** provides an exhaustive list of all dynamic calls resolutions that are identified within the selected project. Furthermore, it allows the user to check and override if needed the existing resolutions. The result of the change can then be seen when the corresponding graph is generated or reloaded.

To access the **Manage Resolutions** analysis tab, in **Application Discovery Browser** perspective, from the Projects list in the central part of the Explore Projects view, select your project. For details on how to select a perspective and a description of **Application Discovery Browser** perspective, see the section AD Perspectives Common Elements.

In the Explore section to the right of the Explore Projects view, double-click Manage Resolutions, the **Manage Resolutions Analysis** tab is displayed as in the following image.

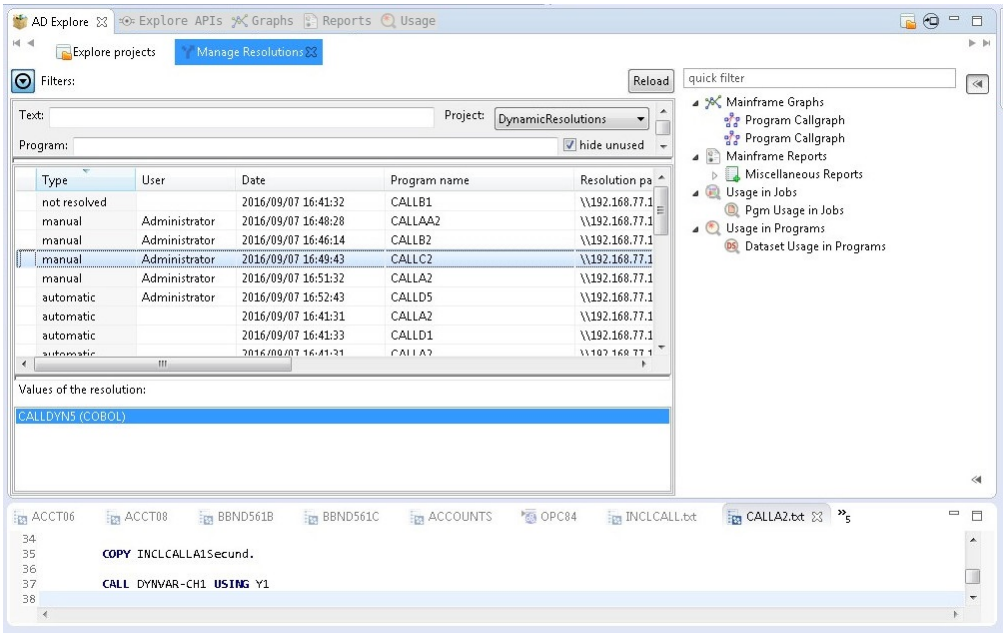


Figure 36. Manage Resolutions Analysis Tab

The central part of **Manage Resolutions** analysis tab displays a list of all dynamic calls resolutions that are identified in the selected project. For each resolution the following information is available:

- The resolution type (either automatic, manual, or not resolved),

- The user who provided the resolution.
- The date and time when the resolution was provided.
- The name of the program that contains the dynamic call for which a resolution was provided.
- The resolution path (the path to the call statement), that is, the path to the program or include that contains the dynamic call for which a resolution was provided.
- The number of the line where the dynamic call appears.
- The statement type,
- The absolute program path (in case the call is made within an include file, the path to the program where the include is used).

The value set for a resolution is displayed in the Values of the resolution area once it is selected from the resolutions list. Selecting a resolution value, that is, a program name from the Values of the resolution area, displays in the area to the right of the tab all the analyses that can be generated for that program.

Use the fields in the Filters area to filter the resolution list. You can filter by text, by program name, by resolution type, or by project; real-time filtering is available. The option **Hide unused** is selected by default and hides manual resolutions that were not applied at the time of the latest build. Unused resolutions are existing resolutions that point to lines that no longer contain a dynamic call. If this option is not selected, the unused resolutions appear in the resolutions list marked with a warning icon. For these resolutions, special actions are available on right-click: **Delete** (removes it) or **Reassign resolution** (the user can decide to reassign values of this resolution to another automatic resolution present in that program).

When the filtering operation is finished, you can use the black arrow button in the upper left corner of **Manage Resolutions** analysis tab to hide the filters and maximize the Results area.

When you identified the resolution that you want to check or modify, select it from the list and then right-click to display the pop-up menu. Two options are available: **View Source** and **Override resolution**. Selecting **View Source** opens the source file in the Editor. Selecting **Override resolution** displays the **Override Current Resolution** window.

Override Current Resolution
Override Resolution allows the user to change the existing resolution for a dynamic call.

To access the **Override Current Resolution** dialog window:

- From a graph (Job Call graph, Program call graph, Program flow graph, Screen call graph, and Transaction call graph), right-click on a red arrow edge (representing a dynamic call) and from the pop-up menu select **Override Resolution**.
- From the **Graph Inventory** analysis tab, click **Resolutions** to display the resolutions list, select a resolution, right-click and from the pop-up menu select **Override Resolution**.
- From the **Manage Resolutions** analysis tab, select a resolution from the list, right-click and from the pop-up menu select **Override Resolution**.

The Override Resolution dialog window is displayed as in the following image.

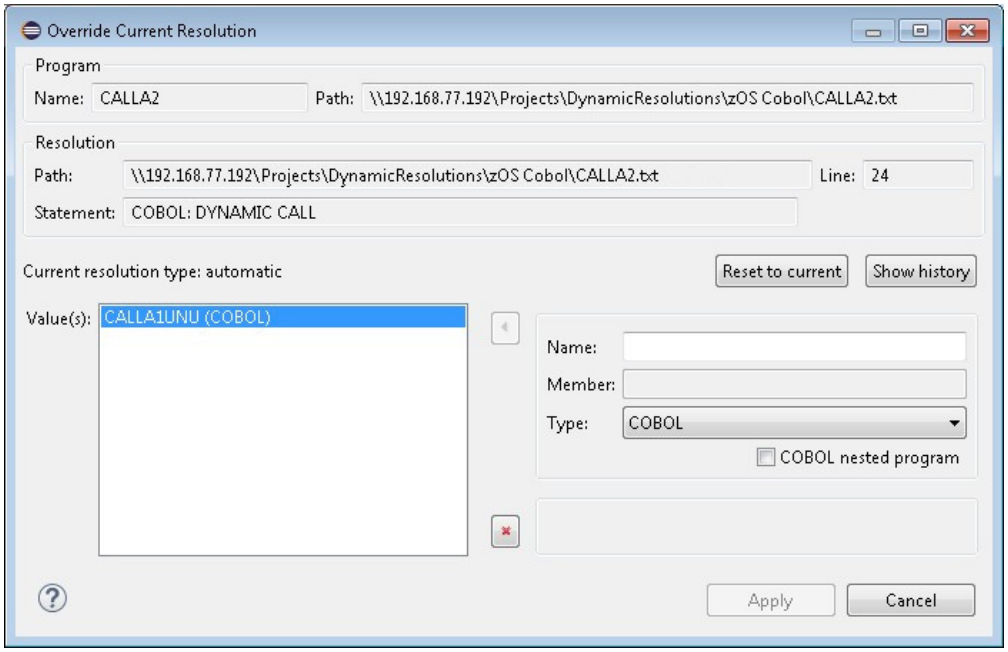


Figure 37. ***Override Current Resolution***

The name of the program that contains the dynamic call and the absolute path to that program are displayed in the upper part of the dialog.

The details of the current resolution for the dynamic call are displayed in the Resolution area of the dialog window: resolution path (in case the dynamic call is made within an include file), the line number, and the statement type.

The current type of the resolution is displayed (automatic, manual, or not resolved) as well as the current resolution value or values.

To change the current value of a resolution, use the fields in the lower right corner of the dialog window, enter the new called program Name, the Member name (only applicable for ASSEMBLER_CSECT and ASSEMBLER_ENTRY program types), and the program type. Then, use the arrow button to the left of the **Name** field to add the new value to the list. You can use **Delete** to delete a value from the list.

The check-box **COBOL nested program** is used to mark a nested program (nested programs can also be called). Use this option only if the program added in the **Name** field is contained within the program where the resolution is overridden. Once a nested program is added to the Values list, if the option COBOL nested program was selected, the name of the parent and the name of the child programs are displayed as a single entry in the list. After you finish editing of the values list, make sure to click **Apply** to save the modifications.

To view a list of resolutions previously applied at the same place as the current one, click **Show History**. A dialog window shows all the previous manual and automatic resolutions only after a build selection was performed. If a build project is performed only the current resolution is shown.

To reapply one of the previous resolutions, select it from the list then click **Choose** to close the dialog window.

Back to the main Override Resolution dialog window, use **Reset to current** to go back to the resolution existing when you opened the Override Resolution dialog.

Back to the resolution list in the **Manage Resolutions** analysis tab, click **Reload** to display the updated resolution in the list.

You can then generate a new graph to display the result of the resolution change or reload an existing graph.

Explore ... (project name) Tab

Explore ... (project name) tab allows the user to explore the project contents and to see the list of available project resources.

Note: This function is only available for mainframe projects.

The resource types available in the Resource type drop-down list depend on the type of project that was created in IBM AD Build.

To display the project contents, in **Application Discovery Browser** perspective, from the Projects list in the central part of the Explore Projects view, select your project. For details on how to select a perspective and a description of Application Discovery Browser perspective, see the section AD Perspectives Common Elements.

In the Explore section, to the right of the Explore Projects view double-click Explore project. The **Explore ...(project name)** tab is displayed as in the following image.

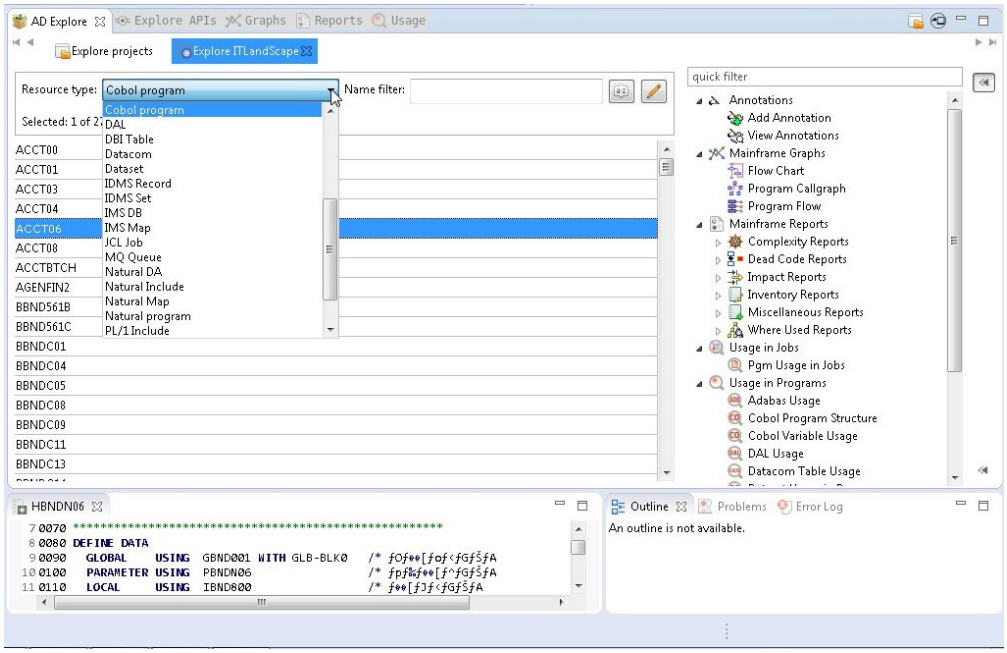


Figure 38. Explore Project tab

The **Resource type** field allows the user to filter the resources that are displayed in the central part of the screen according to the resource type. Use the **Name** field to filter the resources by name. Use the a-z button to the right of the **Name filter** field to order the list alphabetically. Click the adjacent button **display only annotated** to filter the search results so that only the annotated resources are shown. The list of items contains both resources that were physically added to the project and resources that exist in the database but are not physically added in the project. For example, programs that are not physically added but are called by other programs that exist in the project).

Select a resource from the list: the analyses pane to the right of the screen displays the analyses that can be generated for the selected resource. Double-click a resource from the list to open it in the Editor view in the lower part of the screen.

When a resource is selected, the Properties view displays details about the selected resource such as the resource type and its path. For the resources that do not actually exist in the project - which are only called or used by existing resources, the path is not available. This method can be used to identify this type of resources.

Important: PL/I Programs have special treatment in **AD Analyze**: external procedures from a PL/I file are treated as programs. Therefore, all the analysis show the procedure name prefixed with the member name (example: **member:procedure**) as is the case of **Explore**. To filter the content of the PL/I programs list, special characters can be used. For example, you can use the special character ***** and add to it either

prefixes or suffixes as in **member*** or ***procedure**. In order to filter right after the name displayed in the list, you can use the **member:procedure** notation.

Search in Files Analysis Tab
Search in Files option allows the user to search for specific terms throughout the files of the selected project.

Note: This function is only available for mainframe projects.

Note: VERY IMPORTANT! The search can be performed only if a project was indexed with **IBM AD Batch Server**.

In **Application Discovery Browser** perspective, from the Projects list in the central part of the Explore Projects view, select your project. For details on how to select a perspective and a description of **Application Discovery Browser** perspective, see AD Perspectives Common Elements.

In the Explore section, to the right of Explore Projects view double-click Search in Files: the **Search in Files** analysis tab is displayed as in the following image.

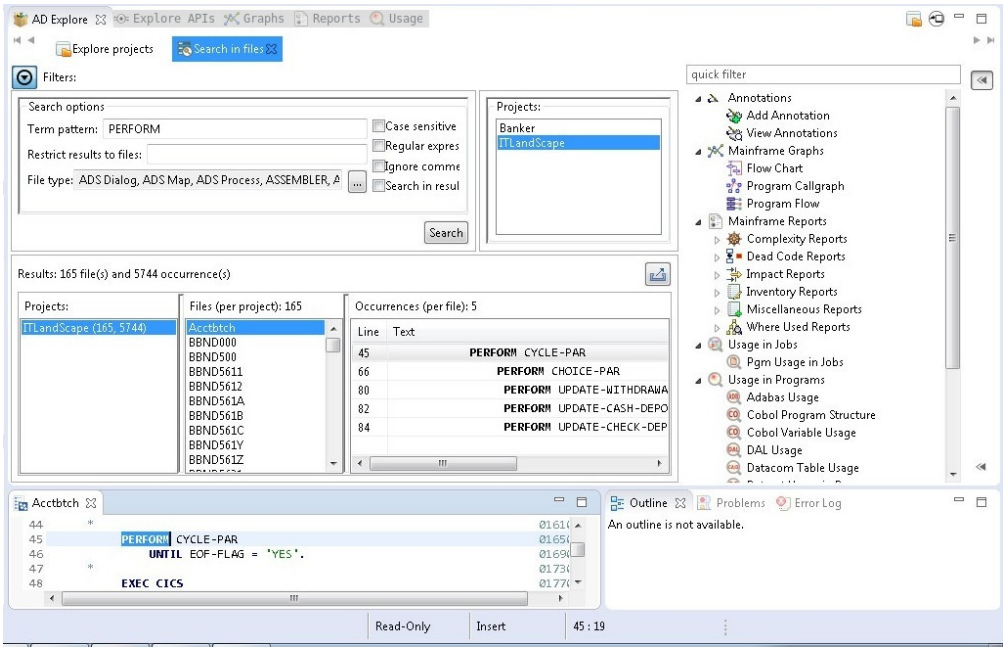


Figure 39. Search in Files Analysis Tab

The following search criteria are available in the Filters area:

Term pattern, enter the pattern that you are looking for. Supported patterns include "a* b*", "a*, b*", "a-b", "a b-c-d", "a @", "a #", "a $", "a?b".

Syntax details.

- "," (comma) means "OR": searching for "termA,termB" shows occurrences (and files) that match any of the terms.
- " " (space) means consecutive terms: searching for "termA termB" shows only occurrences that match both terms in this exact order. Space or spaces right after comma is(are) ignored and multiple spaces are trimmed - considered as one.
- "?" (question mark) means any one character: searching for "abc?" shows occurrences of all four character terms where the first 3 characters are "abc".
- * (asterisk) means any term or any number of characters part of a term. (Because the search is performed on indexed terms and not on full text and a term is considered any sequence of valid characters delimited by separators. For details about which characters are considered as a valid part of a term (and not a separator, and consequently ignored), see https://docs.oracle.com/javase/7/docs/api/java/lang/Character.html#isJavaIdentifierPart(char). To that list the following characters are added:

"@","-" and "#". Characters such as "*", "(", ")", "," (comma), ":", "=", and so on are ignored (considered separators).

These search patterns can be combined to search for the wanted terms.

Restrict results to files, enter the names of the files where you want to perform the search.

File type, select the file types where the search is to be performed. Click the adjacent button to display a list of available file types.

Case sensitive, select this option to make the search case sensitive.

Regular expression, select this option to search for regular expressions. If this option is not selected, the field **Term pattern** can take any text.

Syntax details.

- "," (comma) means "OR", the same as in the syntax list under **Term pattern**. ("regEx1,regex2" is equivalent to "regEx1|regex2").
- " " (space) means consecutive terms, the same as in the syntax list under **Term pattern**.

Note: When Regular expressions option is checked, regular expression matching is also done for the file name.

Note: In order to obtain a result, the search term (both simple text and regular expression) must fully match an indexed term. In regular expression context, this means that if you use, for example, something like an online regex validator/tester for the regex "term", you match "term" plus any word that contains "term" ("terms", "terminates"). When Search in files function is used with "Regular expression" option selected, searching for "term" shows only occurrences of "term". To obtain the same behavior, you must use a regular expression such as "term\w*".

Ignore comment, use this option to exclude comments from the search results.

Note: The following option is only available for Cobol resources.

Search in results, use this option to perform a search within the results of a previous search operation.

The available projects are listed in the pane to the right of the tab. Select the project where the search is to be performed. Once you selected the wanted criteria, click **Search** to perform the actual search operation. When the filtering operation is finished, you can use the black arrow button in the upper left corner of **Search in files** tab to hide the filters and maximize the Results area. The results of the search are displayed in the lower part of the tab and are grouped by project.

Specific analyses can be generated for the files in the Files per project area. Select a file from the list and the available analyses are displayed in the area to the right of **Search in files** tab. Double-click a file from the Files per project area to open it in the appropriate editor. Also, if you double-click an occurrence in the Occurrences (per file) area the corresponding file is opened in the editor at the line where the searched word was found.

To export the results of the Search in files operation, use **Export results to CSV file** in the upper corner of the Results area. The *.CSV file includes all the search criteria used, as well as the number of occurrences per file.

Cobol Source Analysis Tab

Cobol Source Analysis tab provides an inventory of the code elements that are identified in the selected code section.

Note: This function is only available for mainframe projects.

To start the **Cobol Source Analysis**, go to the Editor window, select part of a Cobol source code, and from the right-click menu choose **Cobol Source Analysis**. The corresponding tab is displayed as in the following image.

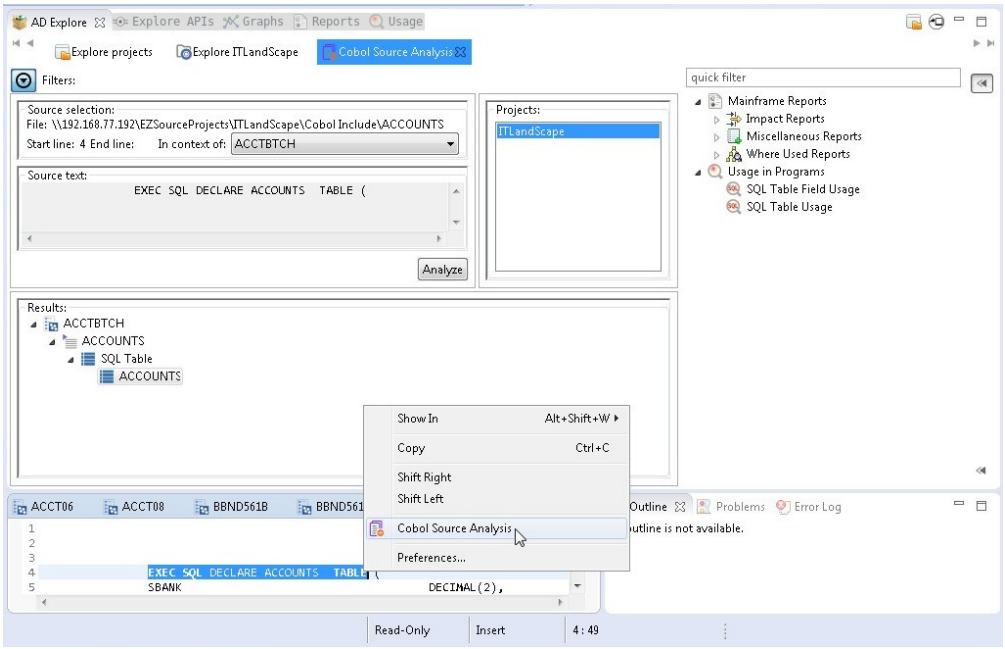


Figure 40. Cobol Source Analysis tab

The Source selection area displays the absolute path to the resource file that contains the selected Cobol source code section. The start line number and end line number of the selection are displayed. The field **In the context of** displays the name of the ancestor program file if the selection was made within the Cobol Include file.

The selected section of Cobol source code is displayed in the Source text area. Click **Analyze** to analyze the selected Cobol source code. The results of the analysis are displayed in the Results area of the tab.

Once the analysis of the Cobol source code is finished, you can use the black arrow button in the upper left corner of Cobol Source Analysis tab to hide the Filters area and maximize the Results area.

The results of the analysis consist of a list of source code elements that are identified within the selected Cobol source code section. The results are organized in a tree structure where the root is the Cobol file and the identified elements are grouped in categories such as Includes, Variables. Click an element from the list. In the Explore area to the right of **AD Explore** tab the analyses that can be generated for the selected element are displayed.

View Expanded Source
View Expanded Source allows the user to view the content of preprocessed COBOL and PL/I programs or JCL.

Note: This option is only available in mainframe projects for Cobol, PL/I and JCL resources if the preprocessing feature was configured in **IBM AD Batch Server**. For details, see `IBM AD Batch Server Configuration Guide`.

The COBOL preprocessing implies expanding all include files and applying all REPLACE compiler directives. For JCLs, preprocessing implies expanding all JCL procedures and JCL include files.

For PL/I , preprocessing implies expanding all include files, with the following supported syntax:

```
File inclusion: %INCLUDE File, %XINCLUDE File, ++INCLUDE File, -INC File
```

```
Library member inclusion: %INCLUDE MEMBER, %INCLUDE LIB (MEMBER)
```

Cobol Resource Expanded

To view the Cobol expanded source, go to the Editor window and from the right-click menu choose **View expanded source**. A new tab is displayed indicating that it contains the expanded resource.

The expanded Cobol code contains a comment that is inserted before a copy expansion, full path of the Copy file used with the source line number for the Copy/Include statement, and a comment inserted at the end of the expansion with the same copy file information (path and line number) matching the start comment.

Example 1 REPLACE compiler directing statement

REPLACE Compiler directing statement replaces one or more words with another sequence of words (or by nothing), in all sources, including all nested (copied) ones, until REPLACE OFF statement is met or till the end of the program if REPLACE OFF is not specified.

Note that a blank space in the text to be replaced may mean one or more blanks in the source itself. In the following example, two pseudo texts were replaced with two other.

REPLACE compiler directing statement:

```
REPLACE ==TEXT TO BE REPLACED== BY ==NEW TEXT==
==EXHIBIT NAMED== BY ==DISPLAY==
```

Example 2 COPY statement replacing

COPY statement can be found in a basic format, copying only the source code as is without changes (unless REPLACE is active at the point of COPY), or with a REPLACING clause which can modify the text within that COPY and all copies nested within it.

A blank in the text to be replaced can translate into one or more blanks in the source itself, for example, two token replacements and one pseudocode replacement.

COPY statement replacing:

```
COPY CPY00001 REPLACING V1 BY V0001
                                V2 BY V0002
==TEXT TO BE REPLACED== BY ==NEW TEXT==
```

Example 3 "-INC" replacing

"-INC" is another format of Copy, and copies only the source code as is without changes unless used with prefix.

The additional information for performing the replacement is stored into the repository.

Example 3 Prefix replacing

Prefixes are used together with "–INC" commands and are defined within comments in the original Cobol source.

After a prefix is defined, each record or variable name that is mentioned in variable definition will have a prefix as indicated in the prefix declaration, including a hyphen. The prefix is added to all variables defined following the Prefix start, these could be both in the program code and copybooks code, until a comment with PREFIX,END is found. Prefix would be applied on all nested COPY/-INC too.

A prefix active in the scope of one "–INC" command.

```
     01 ITF1-AREA.
           *PREFIX,ITF1
-INC CCMNT000
           *PREFIX,END
```

If copybook CCMNT000 has the following definitions:

copybook CCMNT000 definitions:

```
    02  REC1.
      03 V1 PIC 999
      03 V2 REDEFINES V1 PIC 9V99
      03 FILLER PIC XX.
```

The expanded copybook has variables with “ITF1-“ prefix:

expanded CCMNT000 copybook:

```
     02  ITF1-REC1.
       03 ITF1-V1 PIC 999
       03 ITF1-V2 REDEFINES ITF1-V1 PIC 9V99.
       03 FILLER PIC XX.
```

Example 4 SQL Include

Embedded SQL INCLUDE commands are handled in a similar way to a simple COPY statement.

The SQL include files are inserted in the place of the EXEC SQL INCLUDE statements.

JCL Resource Expanded

To view the expanded JCL source, go to the Editor window and from the right-click menu choose View expanded source. A new tab is displayed indicating that it contains the expanded resource.

The JCL source code expansion process consists of the following operations:

- Per each symbol in the Job, performs the replacement;
- Per each Proc/Include usage, performs the symbol replacement and expands the Proc file;
- The Proc file or Include file content is included after the Proc step or Include command and appears with two leading characters (‘XX’) for every line, replacing the original two slash characters used in the PROC/Include (‘//’).

JCL symbols are defined or set in the job code. The symbols are similar to template variables that can be initialized and set along the JCL Jobs Procs and include files code.

When defined, the symbols are used as a name followed by equal sign and value, and when used in the Job code or Proc or Include code, the same name is used with an ampersand ("&") before it. When the Job is expanded, the symbols are replaced both within the job code and Proc/Include code, from the point of definition of the symbol till the end of that job. Symbols are relevant for the Job they are defined in, and other jobs defined in the same JCL file don’t use them. The Proc code may have symbols set at the beginning of the Proc as a default value in case the Job did not override (define or set) some/all symbols before calling the proc.

The reference for symbols is done with "&" before the symbol name and period following it (if there is more text after the symbol) as in the example below.

Original Proc code:

```
//STP010 EXEC PGM=IKJEFT1B,
// REGION=&RGN
//XSBCF DD DSN= &PARML..CONTRLIB(PISYPP20),
// DISP=SHR
//XSOCF DD SYSOUT=*
//PSZP316 DD DSN=&&STP0100,
// DISP=(NEW,PASS),
// DCB=(BLKSIZE=21600,LRECL=120,RECFM=FB),
// SPACE=(&CYL,(5,5),RLSE),
// UNIT=(&WORK,&VOL)
//PSZP317 DD DSN= &DSNGP..P00PS.BP.PSBMPP20,
// DISP=(NEW,CATLG),
// DCB=(BLKSIZE=21600,LRECL=120,RECFM=FB),
// SPACE=(&CYL,(5,5),RLSE),
// UNIT=(&DATA,&VOL)
```

Proc lines after symbol replacement:

```
XXSTP010 EXEC PGM=IKJEFT1B,
XX REGION=4096K
XXXSBCF DD DSN= PJ.OPRPO.CONTRLIB(PISYPP20),
XX DISP=SHR
XXXSOCF DD SYSOUT=*
XXPSZP316 DD DSN=&&STP0100,
XX DISP=(NEW,PASS),
XX DCB=(BLKSIZE=21600,LRECL=120,RECFM=FB),
XX SPACE=(CYL,(5,5),RLSE),
XX UNIT=(WORK,6)
```

```
XXPSZP317 DD DSN= PJ.P00PS.BP.PSBMPP20,
XX DISP=(NEW,CATLG),
XX DCB=(BLKSIZE=21600,LRECL=120,RECFM=FB),
XX SPACE=(CYL,(5,5),RLSE),
XX UNIT=(DATA,6)
```

Double ampersands before a word (for example, "&&STP0100") are not a symbol reference but rather a temporary file name.

PL/I Resource Expanded

To view the PL/I expanded source, **Editor window** > **right click** > **View Expanded Source**. A new tab is displayed indicating that it contains the expanded resource. The expanded PL/I code contains the following information:

- a comment that is inserted before an include expansion.
- the full path of the Include file used with the source line number for the Include statement.
- a comment inserted at the end of the expansion with the same include file information (path and line number) matching the start comment.

Example *%INCLUDE* replacing

The "*%INCLUDE*" preprocessor statement is used to include the text of another file, having the effect of including the specified file code without any further changes.

AD Graphs Tab

Graphical Analysis Overview
An overview of the graphical analysis features available in AD Graphs tab from the **Application Discovery Browser** perspective of **IBM AD Analyze Client**.

The graphical analysis results are presented in **AD Graphs** tab. This view is the main analysis view that displays the results of the analysis that was performed on the selected project or project component. For example, the following picture shows a Program call graph displayed in the **AD Graphs** tab.

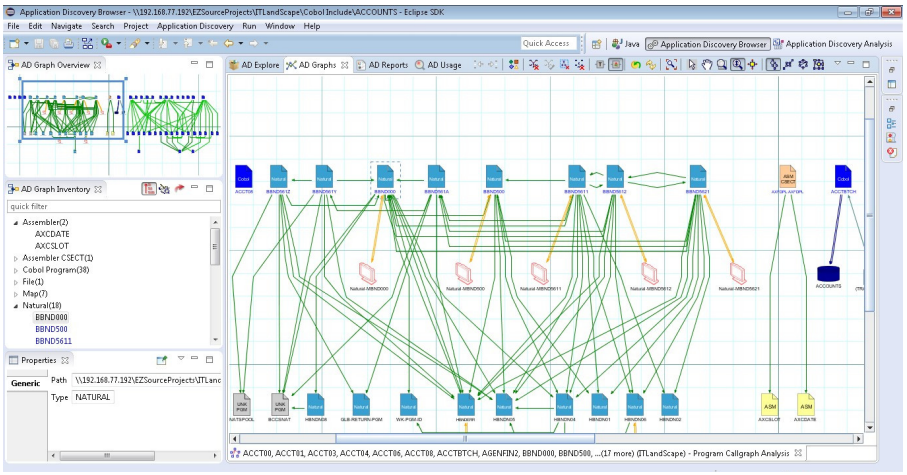


Figure 41. AD Graphs tab

The graphical analysis is presented in three views:

- AD Graphs - the main view, in the central part of the screen - displays the graphs that are generated in **IBM AD Analyze Client** for the selected project. The graphs are presented at a zoom of 100%. The nodes in the graph represent the project components, while the edges (arrows) represent relationships between the components. Both nodes and edges are color coded to show the type of component or relationship. To see a description of all the nodes and edges available in the currently displayed graph, click **Show Graph Legend** from “Graphical Analysis Toolbar” on page 322. The graph legend is presented on the right side of the screen. For most graphs, you can use the symbols in the graph

legend to show/hide the nodes and edges of the graph. The graph symbols depend on the type of graph displayed.

- **AD Graph Overview** view - presents a zoomed-out version of the graph displayed in AD Graphs view so that the entire graph is available at a glance. For details, see AD Graph Overview Area.
- **AD Graph Inventory** view - shows a hierarchical, tree-like list of all the components in the current graph. Some components have specific icons, for details see **AD Graph Inventory** View.

Several types of graphs can be generated depending on the type of project or on the number of selected projects, for details, see Graphs chapter.

Various parts of the graph can be hidden and redisplayed by using commands on pop-up menus, and also by positioning the selection box in the **AD Graph Overview** view. Other information can be obtained by double-clicking components in the graph (which usually causes the corresponding source file to open in an Editor window), or by positioning the mouse cursor over an element.

For more information about each type of graph, see Mainframe Graphs and Java Graphs.

Export Paths

As Mainframe applications are often complex and monolithic, you can use IBM AD to cut large applications into smaller ones to support the Mainframe applications transformation process. Circular layout of AD graphs is helpful to visualize consistent subprojects of large applications.

By right-clicking in the **AD Graphs** tab, you can use the **Export Paths** button to:

- Export the paths for all programs and their dependencies that were used to generate the graph.
- Export the paths for one or more programs and their dependencies.

Note: The **Export Paths** option is not available in the **AD Graphs** tab if the **Expand Callgraph using API to other projects** check box is selected in **IBM AD Analyze Client** > **Preferences** tab. Also, the **Export Paths** option is not available for **Cross Applications Callgraph** and **Cross Applications Callgraph API**.

The graph is exported as a CSV file. Use **Browse** to select the location where you want to export the graph and click **Save** to begin the export operation.

A pop-up message, which informs that the export was **done** is displayed and it asks if you want to open the exported file.

The CSV file format is as it follows:

```
SOURCE_TYPE;SOURCE_NAME;SOURCE_PATH
```

Example of a CSV file when **Program Callgraph** is performed on a **COBOL program**:

SOURCE_TYPE;SOURCE_NAME;SOURCE_PATH
COBOL;ACCT04;C:\ad\Projects\ProjGIT\zOS Cobol\Acct04
COBOL;ACCT05;C:\ad\Projects\ProjGIT\zOS Cobol\Acct05
COBOL_INCLUDE;ACCOUNTS;C:\ad\Projects\ProjGIT\Cobol Include\ACCOUNTS
COBOL_INCLUDE;ACCTCODE;C:\ad\Projects\ProjGIT\Cobol Include\ACCTCODE
COBOL_INCLUDE;ACCTERR;C:\ad\Projects\ProjGIT\Cobol Include\ACCTERR
COBOL_INCLUDE;ACCTMNU;C:\ad\Projects\ProjGIT\Cobol Include\ACCTMNU
COBOL_INCLUDE;ACCTRAN;C:\ad\Projects\ProjGIT\Cobol Include\ACCTRAN

*Figure 42. **CSV** file format*

Graphical Analysis Toolbar
Graphical analysis toolbar from **AD Graphs** tab is presented in detail.

The following buttons are available in the graphical analysis toolbar:

Table 11. Graphical Analysis toolbar buttons	
Function	**Description**
Move between windows	When multiple windows are open, enables the user to navigate back and forward between the open windows.
Legend	Displays the legend for all possible component types in the graph, for example: flow junctions, decision nodes, and so on.
Hide all edges	Hides all edges in the currently displayed graph.
Show all edges	Shows all edges in the currently displayed graph.
Hide selected	Hides the selected graph element.
Hide unconnected	Hides the unconnected elements.
Reload content	Reloads the currently displayed graph.
Refresh contents	Refreshes the drawing.
Fit in canvas	Restores image of entire graph after zooming.
Pointer	Allows the user to select a component and drag and drop it.
Panning	Enables the user to move around the graph.
Marquee zoom tool	Defines area to concentrate on (zoom). This function is helpful if you know which area of the graph you want to concentrate on.
Interactive zoom tool	Enables the user to zoom in and out by moving the mouse: Point to a component with the mouse, hold down the left mouse button. If you move the mouse down (towards you), the graph gets bigger, if you move up the mouse (away from you), the graph gets smaller. This function is useful, for example, for reading text on a graph.
Navigate edges	Enables the user to navigate across edges within the graph. If you click **Navigate edges**and then right-click on a component, it iterates between all of the edges that are connected to the component (meaning, which arrow leaves or enters the current node), showing the name of the destination node each time as a hint. Left-clicking moves the graph along the selected edge. IMPORTANT: Refrain from moving the mouse when you are using this tool, it can lose focus.
Refresh layout	Enables the user to select a symmetrical, circular, or orthogonal layout for the graph. The default layout is hierarchical.

AD Graph Overview Window

In **AD Graph Overview** window the entire graph is presented, and a zoom square marks the current area that is shown in the main tab.

This view is used to select a specific area of the graph that is displayed in the **AD Graphs** tab and to navigate around the graph. This function is useful when the graph is large and cannot be seen in its entirety at any one time.

To select the area of the graph that you want to display in the **AD Graphs** tab, draw a rectangle with your mouse in **AD Graph Overview** window and drag it to mark the required section of the graph.

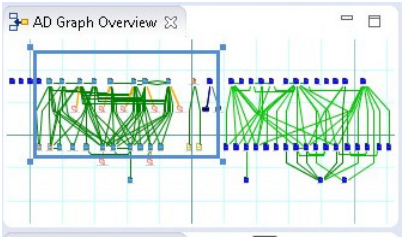


Figure 43. AD Graph Overview Window

Every time you resize the rectangle, **AD Graphs** tab is updated. Also, changes made in the graph that is displayed in **AD Graphs** tab are reflected in AD Graph Overview window.

Graph Inventory View

AD Graph Inventory view shows a hierarchical tree-like display of all the components in the currently selected graph. By default, this view is displayed in the upper-left corner of the screen. The quick filter field can be used to filter the different components in the graph inventory according to the text typed in that field. Typing some text leaves in the inventory only the items that have that text somewhere within their name.

An example of an AD Graph Inventory view is displayed in the following image.

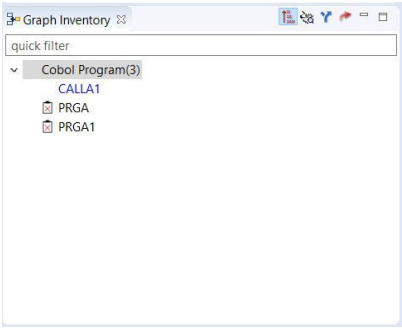


Figure 44. AD Graph Inventory View

In the AD Graph Inventory view are also displayed programs that are not physically added in the project but are called by other programs that are present in the project. If a program has the icon it means that it is not physically added in the project.

By default, all graph elements are listed in the AD Graph Inventory view.

Use **All** button to see all nodes of the graph grouped by their type.

Use **Annotated** button to show only the graph elements with annotations.

Note: Only **Program Call graph**, **Program Flow**, **Screen Call graph**, **Job Call graph**, **Transaction Call** and **API Call Graph** graph analyses support annotations management.

Use **Resolutions** button to show all dynamic calls from that graph, whether resolved by the compiler or overridden through a manual resolution. Possible statuses for a dynamic call showed in the list: dynamic call resolved automatically by the compiler, dynamic call not resolved, dynamic call manually resolved. To modify the resolution status of a dynamic call, right-click on the dynamic call and from the menu select **Override Resolution**. For details on the options available in Override Resolution dialog window, see Override Resolution.

Use **Export Graph Inventory** button in the upper right corner of the **AD Graph Inventory** view to export the current graph inventory in CSV format.

AD Usage Analysis Tab

Note: Usage analysis is only available for mainframe projects.

The **Usage** tab displays a set of parameters that allow the user to determine in what way a certain type of entity (such as SQL table, file, include) is used throughout the resources of the selected project.

In **Application Discovery Browser** perspective, from the Projects list in the central part of the Explore Projects view, select your project. For details on how to select a perspective and a description of **Application Discovery Browser** perspective, see AD Perspectives Common Elements. The pane to the right of the screen presents the types of graphs, reports, and usage analyses that can be generated for the selected project. Double-click a usage analysis to display the **Usage** tab as in the following image.

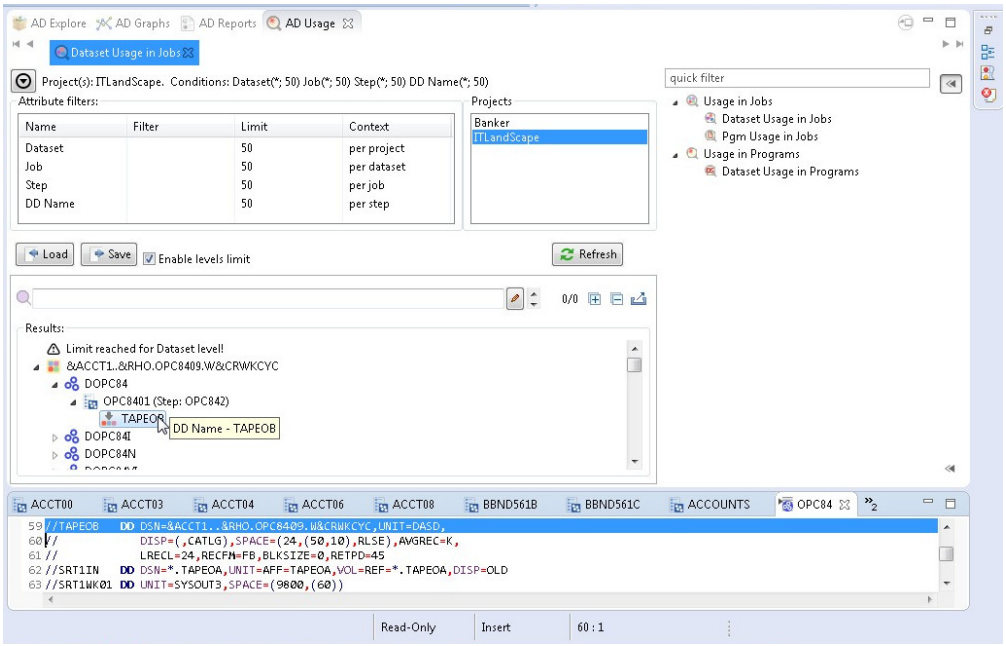


Figure 45. AD Usage tab

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. Only the elements that match the name filter are displayed in the Results area. Special characters can be used; for example, you can use the special character "*" and add to it either prefixes or suffixes as in "*AAA" or "AAA*". To limit the number of attributes of a certain type returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For details, see Usage Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list.

To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements which are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters.

When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the current usage analysis parameters in a separate tab, use the button move to separate view in the upper corner of the tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. Once you select a tree node, the area to the right of the screen is updated and displays the graphs, reports, and usage analyses that can be generated for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The nodes available in the tree depend on the current analysis. For more information on the nodes available for each usage analysis, see “Usage” on page 388.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. For more information on all the Usage analyses that can be generated for each type of project, see “Usage” on page 388.

AD Reports Tab

AD Reports tab displays the report generated for the selected project or resource.

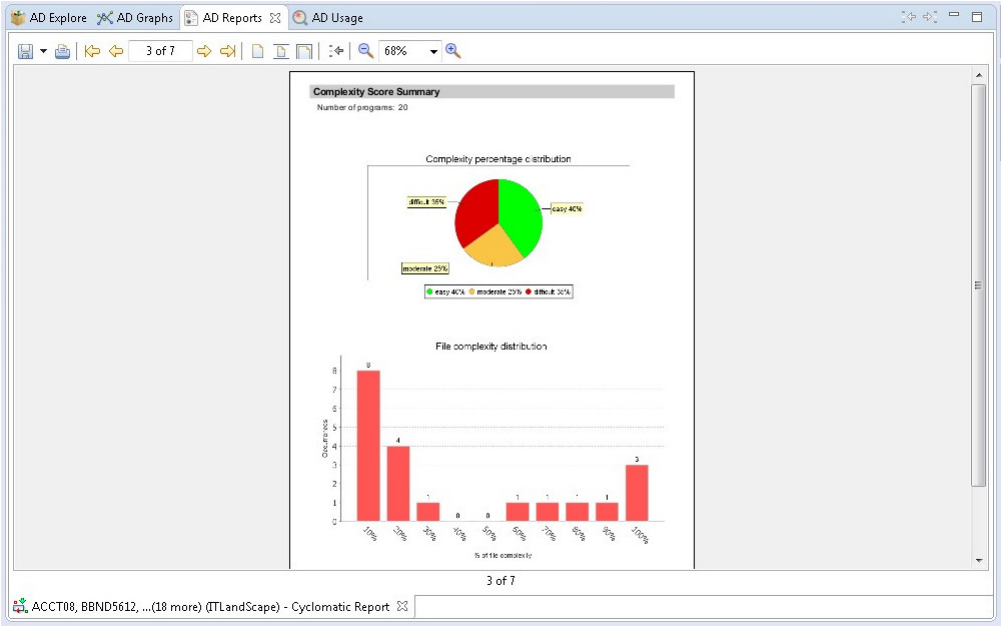


Figure 46. AD Reports tab

Note: For details on how to customize the information that appears on the cover page of the reports, see Reports Settings.

Report toolbar

The toolbar that appears above the reports is used to perform various operations on the report.

The following icons are available.

Table 12. Report Toolbar buttons

Button	Description
Export	Export the report in PDF, RTF, XLSX, DOCX, or CSV format.
Print	Print the report on a printer that is connected to your system.

Table 12. Report Toolbar buttons (continued)	
Button	**Description**
First page	Jump to the first page of the report.
Previous	Go to the previous page of the report.
	Number of the current page.
Next	Go to the next page of the report.
End	Jump to the last page of the report.
Actual size	Present the report in standard size.
Fit page	Present the report to fit to page.
Fit width	Present the report as to fit to width.
Go back	Return to the previous location in the report.
Zoom out	Zoom out (in increments of 25).
	Current® zoom percentage. Select from the drop-down list to zoom to 25% - 200% (in increments of 25).
Zoom in	Zoom in (in increments of 25).

For details on how to generate reports, see the Reports chapter.

Common Elements

Use **AD Perspectives Common Elements** for a detailed description of interface elements common to all **Application Discovery** perspectives.

When launching Eclipse, you can select a perspective, that is, a specific set of views according to the type of project you need to analyze.

IBM AD Analyze perspectives bring together specific **Application Discovery** views and a subset of Eclipse views to create several AD default layouts. You can modify the layout of the selected perspective at any time, by changing the layout of existing views, opening and closing views as required. Some of the views from a perspective are initially docked to the borders of the display area, but they can be undocked and moved, resized, and hidden (closed). Window names are not shown on docked windows. Many of the windows have pop-up menus that are opened by moving the mouse cursor into the window and right-clicking. In some cases, different pop-up menus are available, depending on the exact position of the cursor in the window.

To return to the default perspective layout, right-click on the perspective box at the upper right of the screen and select **Reset** or select **Window** > **Reset** > **Perspective**.

To select one of **IBM AD Analyze Client** default perspectives, follow the instructions on screen and either click the button as indicated or select **Window** > **Open Perspective** > **Other**. A list of default perspectives is displayed.

The following are **IBM AD Analyze Client** specific perspectives:

- **Application Discovery Analysis** and **Application Discovery Flow** - used for the analysis of Java projects.
- **Application Discovery Browser** - used for the analysis of mainframe projects.
- When a perspective is open, the following elements are available on screen.
  - Title bar, displays the name of the current perspective, such as **Application Discovery Browser**.
  - Menu bar, displays the Eclipse and AD Client menus.

- Toolbar, displays the options available for working with Eclipse and AD Client.
- **Explore** tab, in the central part of the screen, displays the projects list if the **Explore projects** tab is displayed, the contents of the selected project if the **Explore project** tab is displayed and the results of a Search in files operation if the **Search in files** tab is displayed. The area to the right of the screen presents the graphs, reports, and usage analyses that can be generated for the selected projects or resources.
- **Explore API** tab, displaying APIs and services downloaded from the AD Catalog web service and defined in z/OS Connect.
- **Graphs**, **AD Reports** and **AD Usage** tabs, in the central part of the screen, display the current analysis output (graph, report, usage analysis) (for details, see AD Graphs Tab, AD Reports Tab, and AD Usage Tab).
- Views, AD Graph Overview, AD Graph Inventory, Properties, Editor, Outline, Problems, Errors - display details or additional information related to the current analysis output.
- Status bar, displays the current system status according to the current task/view.

Some of these elements are part of Eclipse's general set of tools while others are especially created for analysis purposes with **AD Analyze Client**. This section presents a short inventory of the specific **AD Analyze Client** views, for details on the general Eclipse views see Eclipse help files.

Note: The perspective layout can be customized according to the user's needs by moving, resizing, minimizing, maximizing, or closing windows and tabs. Each layout can be saved as a perspective for use in the future. See the Eclipse help files for details on perspectives.

Menu Bar

AD Analyze Client operations are controlled by choosing commands in the Menu bar and pop-up menus, clicking icons on the toolbar, and by using keyboard shortcuts.

The menu bar commands are summarized in the following table. Equivalent keyboard shortcuts, when available, are also listed.

Table 13. IBM AD Analyze Client menus

Menu	Keyboard shortcut	Description
File	Alt+F	Opens the File menu. All options in this menu are standard Eclipse options. For details, see your Eclipse help file.
Edit	Alt+E	Opens the Edit menu. All options in this menu are standard Eclipse options. For details, see your Eclipse help file.
Navigate	Alt+N	Opens the Navigate menu. All options in this menu are standard Eclipse options. For details, see your Eclipse help file.
Search	Alt+A	Opens the Search menu. All options in this menu are standard Eclipse options. For details, see your Eclipse help file.
Project	Alt+P	Opens the Project menu. All options in this menu are standard Eclipse options. For details, see your Eclipse help file.
Application Discovery	Alt+L	Opens AD Client Menu.
Run	Alt+Run	Opens the Run menu. All options in this menu are standard Eclipse options.

Table 13. IBM AD Analyze Client menus (continued)		
Menu	**Keyboard shortcut**	**Description**
Window	Alt+W	Opens the Window menu, which includes both Eclipse and **AD Analyze Client** options.
Help	Alt+H	Opens the Help menu All options in this menu are standard Eclipse options. For details, see your Eclipse help file.

All the menu commands under the File, Edit, Navigate, Search, Project, Run Window and Help menus are standard Eclipse options. For details about these commands, refer to your Eclipse help files.

The options available in Application Discovery menu are described in the following sections.

Application Discovery Menu

According to the version that you purchased, Application Discovery menu can contain several options. One, several or all of the following options might be available at some point in Application Discovery menu: **Annotations**, **Java Reports**, **Export Graph**, **Project Database Connection Test**.

Window Menu

In the **Window** menu, the **Preferences** option offers access to a set of **IBM AD Analyze Client** specific parameters that allow the user to define general, graph, usage, and reports settings and other parameters.

For detailed instructions on how to set these parameters see “Customizing Preferences” on page 429.

Main Screen Toolbar

The main screen toolbar icons enable frequently used menu commands to be performed without having to navigate through the menu hierarchy. A brief explanation of each is presented in the following table.

Note: As the menu can be configured according to the user’s requirements, in this section only the options relevant for **IBM AD Analyze Client** are explained.

Table 14. IBM AD Analyze Main Screen toolbar buttons		
Function	**Menu Bar (Keyboard Shortcut Equivalent)**	**Description**
New	File/New (Ctrl+N)	Starts the New Project/Other wizard.
Save file	File/Save (Ctrl+S)	Saves the current file.
Print	File/Print (Ctrl+P)	Prints the selected/displayed file.
Build All	File/Build All (Ctrl+B)	Builds the current project.
Export Graph	Application Discovery/ Export Graph	Exports the selected graph. For details, see the Exporting the Graph.
Search	Search (Ctrl+H)	Opens the Search window.
Last Edit Location	(Ctrl+Q)	Jumps to the location in the file that was last edited.
Back to	Navigate/Back (Alt+left)	Jumps to the previous file in the list of opened files.
Forward to		Jumps to the next file in the list of opened files.

AD Graph Overview Window

AD Graph Overview window presents an overview of the graph that is displayed in the **AD Graphs** tab; a zoom square marks the current area of the graph that is displayed in **AD Graphs** tab. For details, see **AD Graph Overview** window.

AD Graph Inventory

AD Graph Inventory view shows a hierarchical, tree-like list of all the components in the currently selected graph. For details, see AD Graph Inventory View.

Properties View

The Properties view displays the properties for the selected item, which can be a file, a project, a project node, SQL table, IDMS record/set, and so on. For details, see the Eclipse help.

Text Editor area

AD Analyze Client uses the Eclipse integrated text editor where the content of text files can be viewed and modified. Files that are modified in the Editor are permanently changed.

Note: Files from mainframe projects cannot be modified with the Editor.

To open a file in the text Editor window, select **File > Open** to open the standard Windows Open File dialog box.

Alternatively, double-click a source file icon from the graph to open the corresponding file in an Editor window, often with the appropriate statement highlighted.

For details on the text Editor area, Outline, Problems, and Errors views, see the Eclipse help.

Graphs

Mainframe Graphs and Layouts

A-AUTO Graph

The A-AUTO Graph presents Networks dependencies relationships and Jobs relationship inside a Network or several Networks.

If A-AUTO dataset flag information is available in the project, triggering relationships are also presented in this graph. The A-AUTO dataset flag is a mechanism that allows a Job to trigger an A-AUTO network execution by setting a dataset-related flag, thus linking one network in which this Job was executing to a second network. Networks can reside in the current project or in other mainframe z/OS projects. The A-AUTO dataset flag information is available in a report that is used in the project building process, in **IBM AD Build Client**. For details, see `IBM AD Build Client and Server User Guide`.

To display the A-AUTO graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate an A-AUTO graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click AAUTO Graph. The A-AUTO Graph Analysis wizard is displayed.

Select the starting point of the analysis - job or network - then make your choices in the Graph options area. By default, the graph analysis depth is limited to three levels. The direction of the analysis is forward and the network nodes are collapsed. Modify according to needs, then click **Next** to display the second dialog window of the wizard.

The Available jobs(networks) area displays the resources available for graph generation. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish**.

If the direction selected for the analysis is either backward or both (backward and forward), an extra dialog window is available in the wizard that allows the selection of projects where to look for triggered networks.

Important: This list contains only mainframe z/OS projects. In order to see results, make sure that you select projects that contain the same Dataset flag report file that is used for A-AUTO dataset flag mechanism. See `IBM AD Build Client and Server User Guide` for details on how to include a Dataset flag report in the build process.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

If the direction selected for the analysis is forward, the generated graph has the following characteristics:

- Input networks are expanded by default. The rest of the networks are collapsed; networks that are linked only by triggering relationships cannot be expanded.
- Input networks names are displayed in blue in the graph and in the Graph Inventory window.
- The same icon type is used for all identified networks regardless of whether they reside in the current project or not.

If the direction selected for the analysis is backward, the generated graph has the following characteristics:

- Input networks are expanded by default. The rest of the networks are collapsed.
- In the context of A-AUTO dataset flag mechanism, networks that reside in other projects can be expanded but display only the JCLs that contain the flag that triggered the network.
- Input networks names are displayed in blue in the graph and in the Graph Inventory window.
- Special icons are used to mark the networks that reside in projects other than the current one.
- If more than one project was used as input, in the Graph Inventory window the networks are grouped according to the project where they reside.

By default, a limit icon marks the nodes for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all nodes that reach the level limit set in the wizard are marked with a limit icon regardless of whether further analysis results are available or not. The same criteria apply for displaying the limit icon in the Graph Inventory window.

AD Graph Overview window in the upper left corner of the screen presents an overview of the entire graph, which is useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View. In the Graph Inventory window, a limit icon is displayed in front of the networks on which a depth limitation is set from the wizard.

The Properties window displays the properties of the element selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

A-AUTO graph element pop-up menu

Right-click an element from the A-AUTO Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types: **View analysis source**, **Show properties**, and **Filter**. According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses. For JCL nodes, **View source** option is also available.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

API Call Graph

API Call Graph can be used to visualize details about the implementation of the selected API or service in a certain project and to have a more detailed view of the implementation down to programs, tables or files.

To display the **API Call Graph**, go to **Application Discovery Browser perspective** > **Explore APIs** > **Name list**, and then select the API for which one to generate a API Call Graph.

The **Available Projects** area displays the projects available for graph generation. Select a project, set a callgraph limit then click **Finish** to generate the **API Call Graph**.

CA7 Job Triggering Flow

The CA7 Job Triggering Flow graph shows the scheduling graph starting from the selected job.

To display the CA7 Job Triggering Flow graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a CA7 Job Triggering Flow graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click CA7 Job Triggering Flow to display the CA7 Job Triggering Flow wizard.

The Available CA7 Jobs area displays the CA7 jobs available for graph generation. Use the **Name** field to filter the contents of the list. Select a job from the list then press **Next** to display the second dialog window of the wizard.

The Available Schedule IDs area displays all the schedule IDs available for the CA7 job that was selected in the previous dialog of the wizard. Select one of the available Schedule IDs and the display **Options** then press **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window, in the upper left corner of the screen, presents an overview of the entire graph, which is useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The Properties window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

CA7 Job Definitions

1. **Start job** definition: a job that is not triggered but that triggers at least a job and that is not scheduled.
2. **End job** definition: a job that is triggered but that does not trigger any job, regardless of it being scheduled or not.
3. **Scheduled job** definition: a job that is scheduled regardless of it triggering or not a job.

4. **Middle job** definition: a job that is triggered and that triggers at least a job, regardless of it being scheduled or not
5. The categories of jobs being able to generate a graph are the ones from point 1 to point 4.

CA7 Job Triggering flow graph element pop-up menu

Right-click on an element from the **CA7 Job Triggering Flow** Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element selected from the graph. Here are the options available for all element types: **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

CA7 Paths

The CA7 Paths graph shows the scheduling path between two selected jobs.

To display the CA7 Paths graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a CA7 Paths graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click CA7 Paths to display the CA7 Paths wizard.

The Available CA7 Jobs for first job and Available CA7 Jobs for last job areas display the CA7 jobs available for graph generation. Use the **Name** field at the bottom of the list to filter the contents of a list. Select a start job and an end job from the lists then press **Next** to display the second dialog window of the wizard.

The Available Schedule IDs area displays schedule IDs for the job that was selected as starting point. Select one of the available schedule IDs and the display Options then press **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The Properties window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

CA7 Paths graph element pop-up menu

Right-click on an element from the CA7 Paths graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element selected from the graph. Here are options available for all element types: **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Control-M Graph

AD Control-M Analysis allows the user to display a scheduling graph by choosing the starting job and determining based on that job the condition flow.

For a proper **AD Control-M Analysis** you need to make sure that:

1. The project's `ControlM` folder contains at least one Control-M XML file.

   Note: If there is no Control-M XML file present, you can add it either manually or automatically:

   - If it is previously exported from Control-M Enterprise Manager, you can add it **manually** under the project's folder.
   - It can be **automatically** retrieved from Control-M Enterprise Manager, by using **IBM AD Build Configuration.** For more information, see “Configuring the z/OS Connection” on page 216.
2. The IBM AD project is built. Instructions for performing a Build of the project can be found in topic “Building Projects” on page 264.

To display the **Control-M Graph**, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a **Control-M Graph** graph. For details on perspectives and their components, see “Perspectives” on page 307.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click **Control-M Graph** to display the **Control-M Graph** wizard.

The **Available Control-M jobs** area displays the jobs available for graph generation. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the jobs between the Available and Selected lists. The **Control-M Graph** is generated based on the jobs that are present in the **Selected Control-M jobs** area. Then, select the **Call direction** to see the forward conditional job flow, the backward conditional job flow or both, then click **Finish**.

AD Graphs tab presents the generated **Control-M Graph**. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. (For more information, see Graphical Analysis Toolbar). To see details on the meaning of the elements of the graph, use the **Legend** button. For more information on the graph legend function, see Showing the Graph Legend.

Cross Applications Callgraph API

For cross-project analyses within the mainframe-based projects that follow API calls between projects, use the Cross Applications Callgraph API.

The Cross Applications Callgraph API shows applications and their dependencies without inner control flow. It is similar to the Shared Resources Graph. However, the Cross Applications Callgraph API does not follow or show the sharing of data sources. It is based on control flow, which means calls to programs or transactions, while the Shared Resources Graph is based on data sharing.

Only the mainframe projects from the workspace that has the graph database are considered for the Cross Applications Callgraph API analysis. It can start on one or multiple mainframe projects. The analysis purpose is to compute and display the high-level connections between projects through the following API external calls in a graphical way:

- The call to a program that is matched against the application name.
- The call to a program that is matched against the CICS region name.
- The call to a transaction that is matched against the application name.
- The call to a transaction that is matched against the CICS region name.

Cross Applications Callgraph

Cross Applications Callgraph is a graph that is created to show calls between different mainframe projects that have their databases on the same DB instance.

To have the **Cross Applications Callgraph** available in **IBM AD Analyze Client**, you need to:

- Make sure that **IBM AD Cross Applications Service** is up and running. For more information, see "STEP 9. Configuring IBM AD Cross Applications Service" on page 94 (without authentication) or "STEP 10. Configuring IBM AD Cross Applications Service" on page 145 (with authentication).
- Make sure that the Cross Application feature is **enabled** in **IBM AD Configuration Server** for the specific projects. For more information, see "Enabling Cross Applications" on page 49.
- Perform a full Build of each IBM AD project. Instructions for performing a full Build of the project can be found at "Building Projects" on page 264.

The **Cross Applications Callgraph** can be displayed as follows:

- **At the program level** - the **Cross Applications Callgraph** can be generated at the **program level**.
- **At the project level** - in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select a project, or multiple projects that are built.

In the Mainframe Graphs section, to the right of the *Explore Projects* view, double-click **Cross Applications Callgraph** to display the **Cross Applications Callgraph** wizard.

The *Available projects* area displays all projects that are available for **Cross Applications Callgraph** generation. The *Selected projects* area displays only the projects that were initially selected for the **Cross Applications Callgraph** generation. Use the arrow buttons to move the projects between the *Available and Selected lists*. From the *Call direction* area, select one of the available options: **Forward** or **Backward**. By default, the graph is limited to three levels. If applicable, modify the limit, then click **Finish** to display the **Cross Applications Callgraph**.

Note: If you have only one project available in the *Selected projects* area, you can generate the **Cross Applications Callgraph** at the program level, by clicking **Next**. In the *Available programs* area, all programs that are involved in the calls computation process of different mainframe projects are displayed.

If the **Cross Applications Callgraph** is displayed at the project level, you can choose to expand the internal callgraph, by right-clicking on the target project and by choosing to expand it. As a result, the internal callgraph of the target project is displayed.

Note: Make sure that the projects are imported in the **OrientDB** database. The internal callgraph is generated based on the **OrientDB** database

AD Graphs tab presents the generated **Cross Applications Callgraph**. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. (For more information, see Graphical Analysis Toolbar). To see details on the meaning of the elements of the graph, use the **Legend** button. For more information on the graph legend function, see Showing the Graph Legend.

Dataset Flow

The Dataset Flow graph shows the data flow within a batch process, starting from a physical dataset. The starting dataset can be at the start of the process (forward data flow), at the end of the data flow (backward data flow) or in the middle of it (backward and forward data flow).

To display the Dataset Flow graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a **Dataset Flow** graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section to the right of the Explore Projects view double-click Dataset Flow to display the Data Flow Through Physical Dataset Analysis wizard.

The Available Datasets area displays the datasets available for graph generation. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the datasets between the Available and

Selected lists. By default, the graph analysis depth is not limited and the direction of the analysis is forward. Modify according to needs then click **Next** to display the second dialog window of the wizard.

Select the datasets for which the subsequent tree must be excluded from the graph. The "excluded" datasets are displayed in the graph with a funnel icon that indicates that the subsequent tree was excluded from the graph. Then, press **Finish** to generate the graph.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

By default, a limit icon marks a node beyond which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all nodes that reach the level limit set in the wizard are marked with a limit icon regardless of whether further analysis results are available or not. The same criteria apply for displaying the limit icon in the Graph Inventory window.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The Properties window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Dataset Flow graph element pop-up menu

Right-click on an element from the dataset flow graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Flow Chart

The statement Flow Chart shows the structure of the program on a statement level in flowchart format.

To display the Flow Chart, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a Flow Chart. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section to the right of the Explore Projects view double-click Flow Chart to display the Flow Chart Analysis wizard.

The Available Programs area displays the resources available for graph generation. Use the **Name** field to filter the contents of the list. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Select the resource for which you want to generate the Flow Chart then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. For details, see Graphical Analysis Toolbar.

To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The Properties window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Flowchart element pop-up menu

Right-click on an element from the flowchart to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **Show properties**, and **Filter**. According to the selected element, **Go to source** option can also be available.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Job Call Graph

The Job Call Graph shows the global relationships between a selected JCL and other components of the project.

To display the Job call graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the jobs for which you want to generate a Job call graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click Job call graph to display the Job call graph wizard.

The Available Jobs area displays the resources available for graph generation. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists. By default, the graph is limited to three levels. Modify if applicable then click **Finish**.

AD Graphs tab displays the generated graph. Input nodes are displayed in blue. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

Note: For large, complex graphs you can choose to restrict the number of programs displayed. You can specify a program beyond which subsequent programs must not be displayed. According to the type of database used for the projects, restricted nodes are displayed differently. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

By default, a limit icon marks the nodes for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all nodes reaching the level limit set in the wizard are marked with a limit icon regardless of whether further analysis results are available or not. The same criteria apply for displaying the limit icon in the Graph Inventory window.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

Input nodes are displayed in blue.

Note: In the Graph Inventory window the names of the restricted nodes are displayed differently according to the database used for the projects. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from Preferences window, for details on how to set these restrictions see Call graph restrictions.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Supported resource types for API calls

For the resource types that are supported for API calls, go to section Supported resource types for API calls in topic Program Callgraph.

Job Call graph element pop-up menu

Right-click on an element from the Job call graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element shortcut menu.

Job Flow

The Job Flow graph shows the sequence of steps within the selected job.

To display the Job Flow graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the job for which you want to generate a Job Flow graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click Job Flow to display the Job Flow graph wizard.

The Available Jobs area displays the resources available for graph generation. Use the **Name** field to filter the contents of the list. Select a job from the list then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, click **Legend** . For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details see, AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Job Flow graph element pop-up menu

Right-click on an element from the Job Flow Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **Show properties**, and **Filter**.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Layouts

The Layouts graph allows the user to view the layout of the screen as it appeared in its original system.

To display the Screen Layout, go to Application Discovery Browser perspective, and select **Projects list** > **project** that contains the screen for which you want to see the layout. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the **Layouts** section to the right of the Explore Projects view double-click **Screen Layout**, the Screen Layout wizard is displayed. Select the screen for which you want to see the layout then press **Finish**. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

Mainframe Layout

The Mainframe Layout graph shows the mainframe landscape including CPU, Sysplex, LPAR, installed subsystems (Db2, CICS, MQ, Control-M, IMS, Natural, Adabas, and so on), and devices (Tapes, Printers, Disks).

To display the Mainframe Layout, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project for which you want to generate a Mainframe Layout graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click Mainframe Layout.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The Properties window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Mainframe Layout graph element pop-up menu

Right-click on an element from the Mainframe Layout graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **Show properties** and **Filter**. Use the **Show properties** option to display important information about the Mainframe components in the Properties window.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Program Callgraph

The **program callgraph** shows the global relationships between a selected program and other components of the project.

Note: The graph wizard is presented in detail only in this section. For all the other graphs, only the specific options are presented in the respective sections.

To display the **program callgraph**, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a **program callgraph**. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the **Mainframe Graphs** section, to the right of the Explore Projects view, double-click **Program Callgraph** to display the wizard as in the following image.

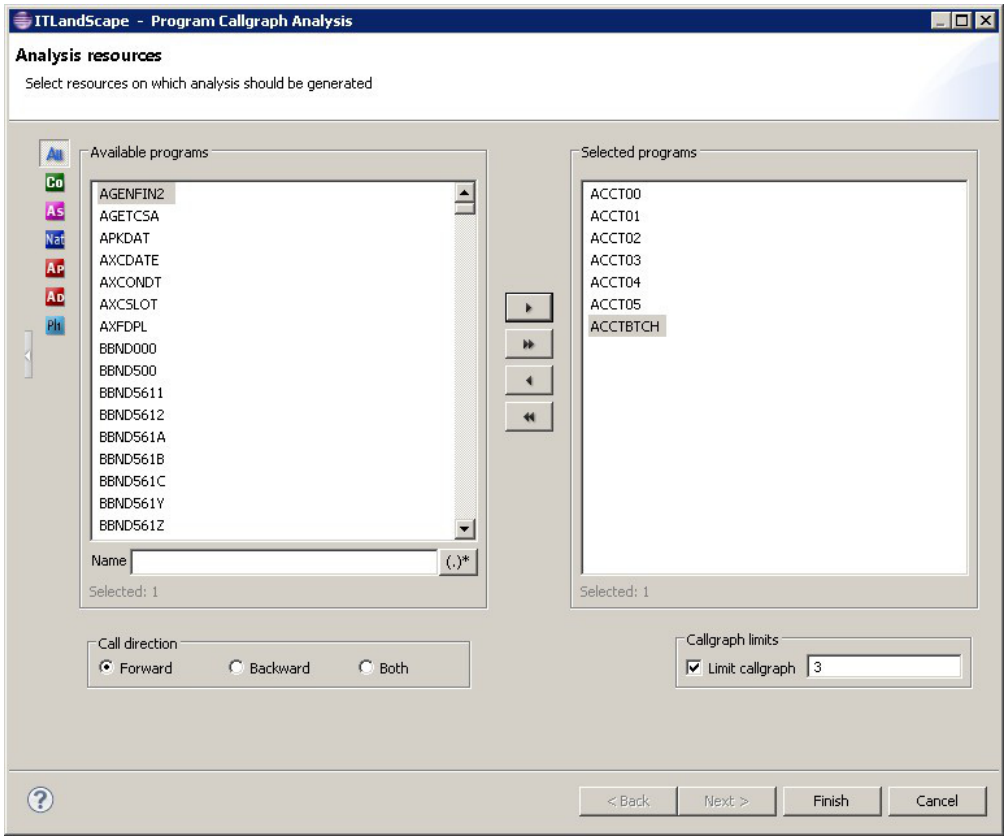


Figure 47. Program callgraph wizard

The Available Programs area displays the resources available for graph generation. To filter the available resources, click one of the buttons to the left of the list to display only certain types of resources. Place the cursor over an icon to display a tooltip about the type of resource it will display and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists. Then select the **Call direction**, that is, whether you want to see the programs and resources that are being called by the program (Forward), or those that call the program (Backward) or both. By default, the graph is limited to three levels. Modify if applicable then click **Next**.

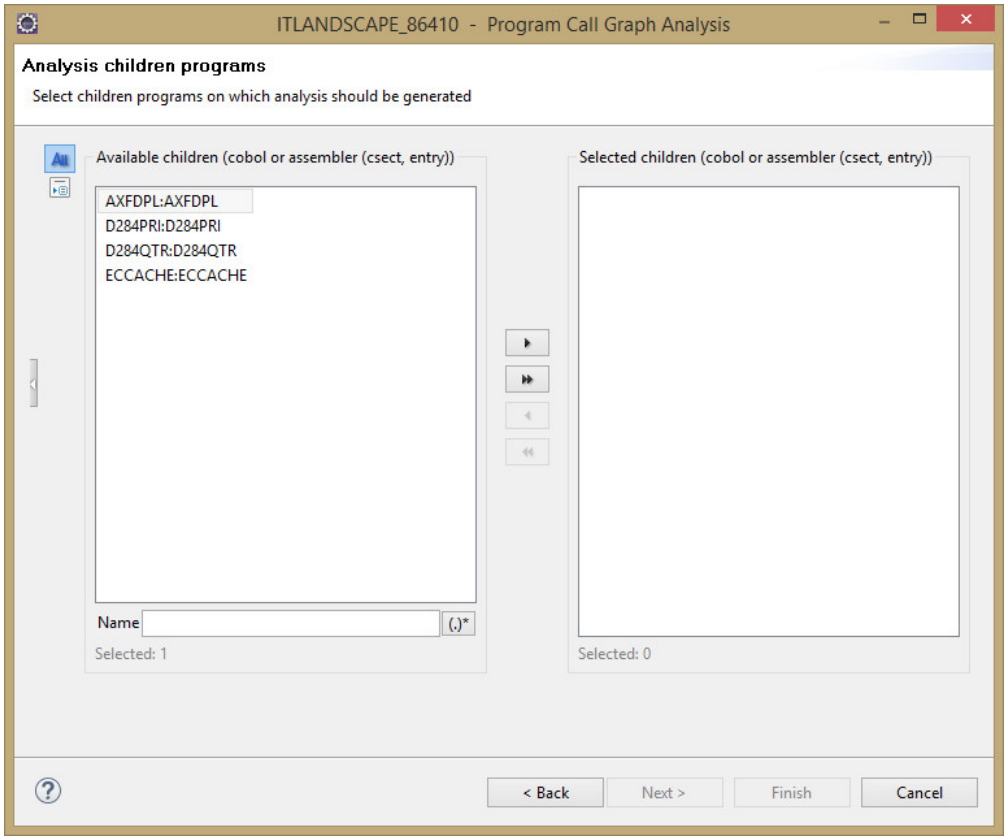


Figure 48. Program callgraph wizard - selecting children

The second dialog window of the wizard displays a list of children available in the selected programs. Select the children that you want to include in the graph then click **Finish** to generate the program callgraph.

AD Graphs tab presents the generated graph. Input nodes are displayed in blue. To navigate the graph, use the buttons in the graph toolbar, which is located in the upper right corner of the screen. (For details see Graphical Analysis Toolbar). To see details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend function, see Showing the Graph Legend.

Note: For large, complex graphs you can choose to restrict the number of programs displayed: you can specify a program beyond which subsequent programs must not be displayed. According to the type of database used for the projects, restricted nodes are displayed differently. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

By default, a limit icon marks the nodes for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all nodes that reach the level limit set in the wizard are marked with a limit icon regardless of whether further analysis results are available or not. The same criteria apply for displaying the limit icon in the Graph Inventory window.

AD Graph Overview window in the upper left corner of the screen presents an overview of the entire graph, which is useful when working with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory view presents a hierarchical list of all the elements in the graph. For details, see AD Graph Inventory View. Input nodes are displayed in blue.

Note:

In the **Graph Inventory** window, the names of the restricted nodes are displayed differently according to the database used for the projects. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

The Properties view displays the properties of the element that is selected from the graph. For details, see Properties View.

Supported resource types for API calls

In a program callgraph, if the extensibility feature is enabled, the following resource types are supported for API calls:

Table 15. Supported resource types for API calls in the program callgraph

Resource type	Statement	Location	Call direction	Expanding to other projects
CICS screen	`CICSReceiveMap` `CICSSendMap`	N/A	Forward Backward Both forward and backward	Not supported
CICS transaction	`CICS Return` `CICS Start`	CICS region Application	Forward Backward Both forward and backward	Supported
COBOL program	`Call`	CICS region Application	Forward Backward Both forward and backward	Supported
File	`CICS File Delete` `CICS File Read` `CICS File Rewrite` `CICS File Write` `File Delete` `File Read` `File Rewrite` `File Write`	N/A	Forward Backward Both forward and backward	Not supported
IMS screen	**`IMSReceiveMap`** IMS function insert (ISRT) **`IMSSendMap`** IMS function get unique (GU)	N/A	Forward Backward Both forward and backward	Not supported

Table 15. Supported resource types for API calls in the program callgraph (continued)

Resource type	Statement	Location	Call direction	Expanding to other projects
IMS segment	**IMSDBDelete** IMS function delete (DLET) **IMSDBInsert** IMS function insert (ISRT) **IMSDBRead** IMS function get unique (GU) **IMSDBReadNext** IMS function get next (GN) **IMSDBUpdate** IMS function replace (REPL)	N/A	Forward Backward Both forward and backward	Not supported
IMS transaction	`IMSStartTransaction`	Application	Forward Backward Both forward and backward	Supported
PL/I program	`Call`	CICS region Application	Forward Backward Both forward and backward	Supported
SQL field	`SQLInsert` `SQLSelect`	N/A	Forward Backward Both forward and backward	Not supported
SQL table	`SQLDelete` `SQLInsert` `SQLSelect` `SQLUpdate`	N/A	Forward Backward Both forward and backward	Not supported

Note: The preceding resource types are also supported for API calls in the job callgraph, screen callgraph, and transaction callgraph.

Graph element pop-up menu

The operations available in the pop-up menu of graph elements depend on the type of element selected. The following options might be available:

- **View analysis source** - opens the analysis source in the Editor window.

  Note: If the selected resource is on a remote system, additional configurations are needed in order to view its contents in the Editor view. For details, see topics “Integrating IBM AD Analyze Client with IDz features ” on page 462 and “Analyzing Resources From an Endevor Remote System” on page 464.
- **Annotations** - shows the Annotations options available for the selected resource.
- **Complexity Reports** - shows the Complexity reports that can be generated for the selected resource.
- **Dead Code Reports** - shows the Dead Code reports that can be generated for the selected resource.

- **Impact Reports** - shows the Impact reports that can be generated for the selected resource.
- **Inventory Reports** - shows the Inventory reports that can be generated for the selected resource.
- **Mainframe Graphs** - shows the graphs that can be generated for the selected resource.
- **Miscellaneous Reports** - shows the Miscellaneous reports that can be generated for the selected resource.
- **Usage in Jobs** - shows the Usage in Jobs analyses that can be generated for the selected resource.
- **Usage in Programs** - shows the Usage in Programs analyses that can be generated for the selected resource.
- **Where Used Reports** - shows the Where Used reports that can be generated for the selected resource.
- **Show properties** - shows the properties of the selected resource in the Properties window.
- **Filter** - shows the filtering criteria available for the selected resource.

Graph edge pop-up menu

Click on an edge from the graph to select it then right-click to display the pop-up menu.

The following options might be available depending on the type of edge selected.

- **Override Resolution** - available only for a red arrow edge (representing a dynamic call) in job callgraph, program callgraph, program flow graph, screen callgraph, and transaction callgraph. Displays the Override Current Resolution dialog. This function is available only for mainframe projects for which a graph database was set up. Currently, the Override Resolution function is implemented only for Cobol and Assembler programs.
- **View analysis source** - opens the analysis source in the Editor window.
- **Show properties** - shows the properties of the selected edge in the Properties window.

Navigating in the graph

In addition to the standard horizontal and vertical scroll bars, the Zoom, Pan, and Resize icons can be used to help move around the graph. Also, you can select a node and drag it to a new location. When a node is moved, all the connectors that are attached to it are changed to fit the new location.

Graph Layout Examples

In this section, an example of each type of graph layout is presented. The same data is presented in each layout type illustration.

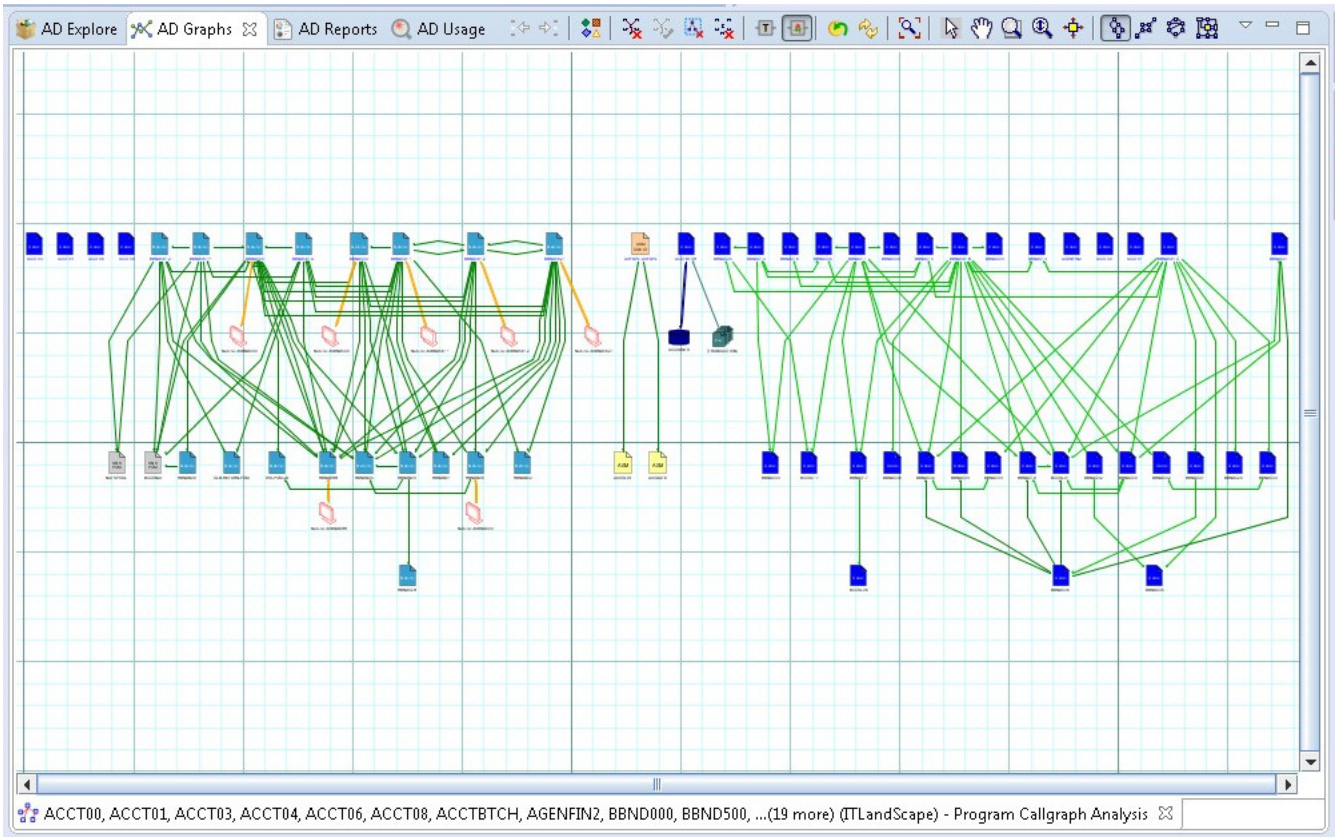


Figure 49. Hierarchical graph layout

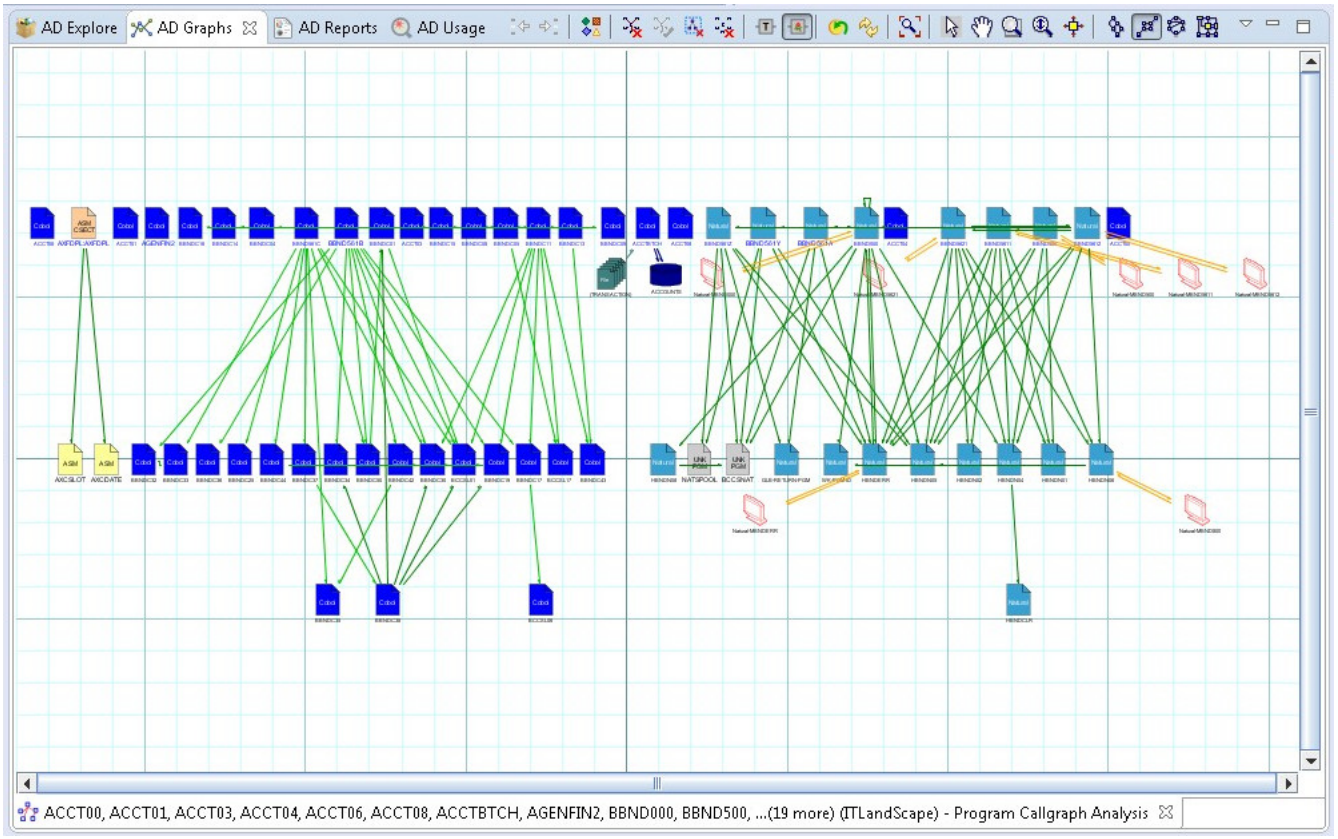


Figure 50. Symmetric graph layout

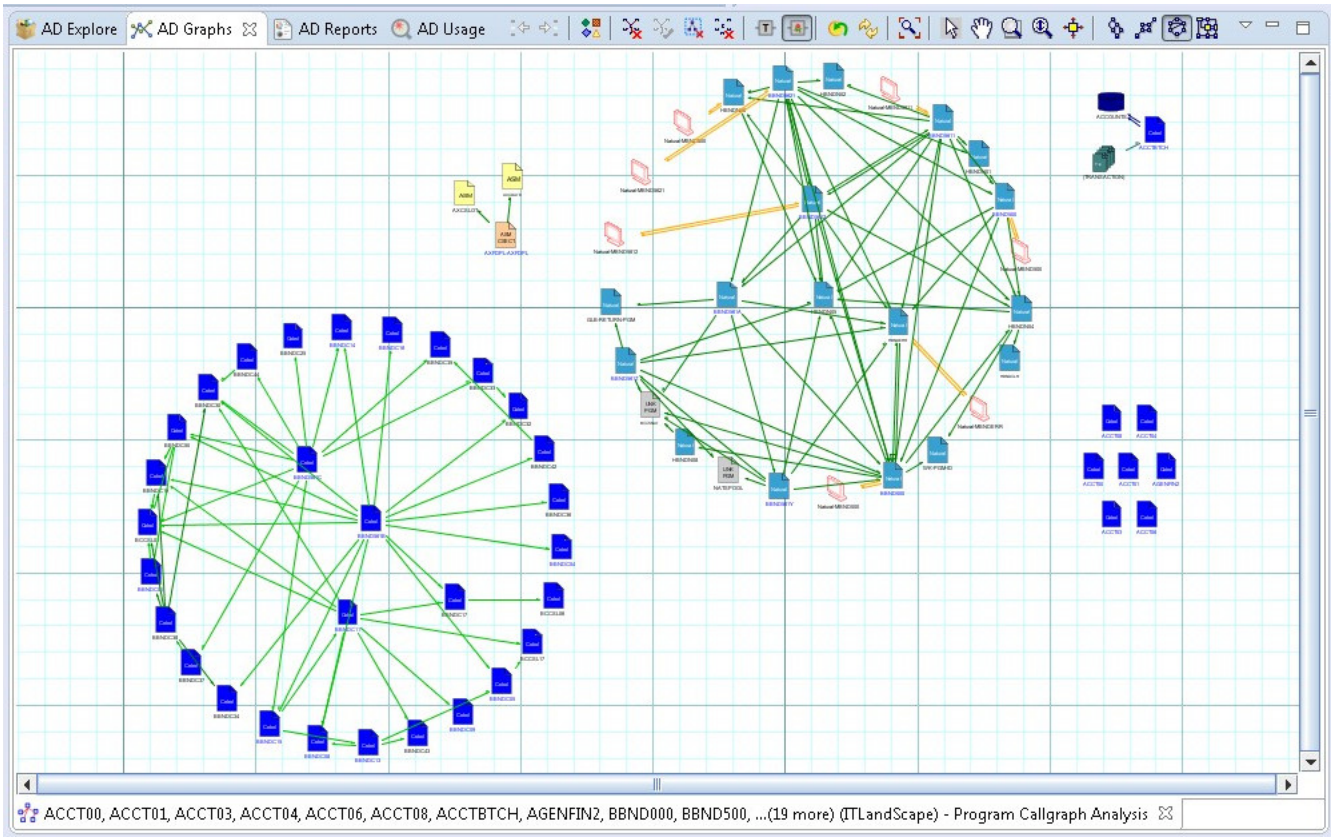


Figure 51. Circular graph layout

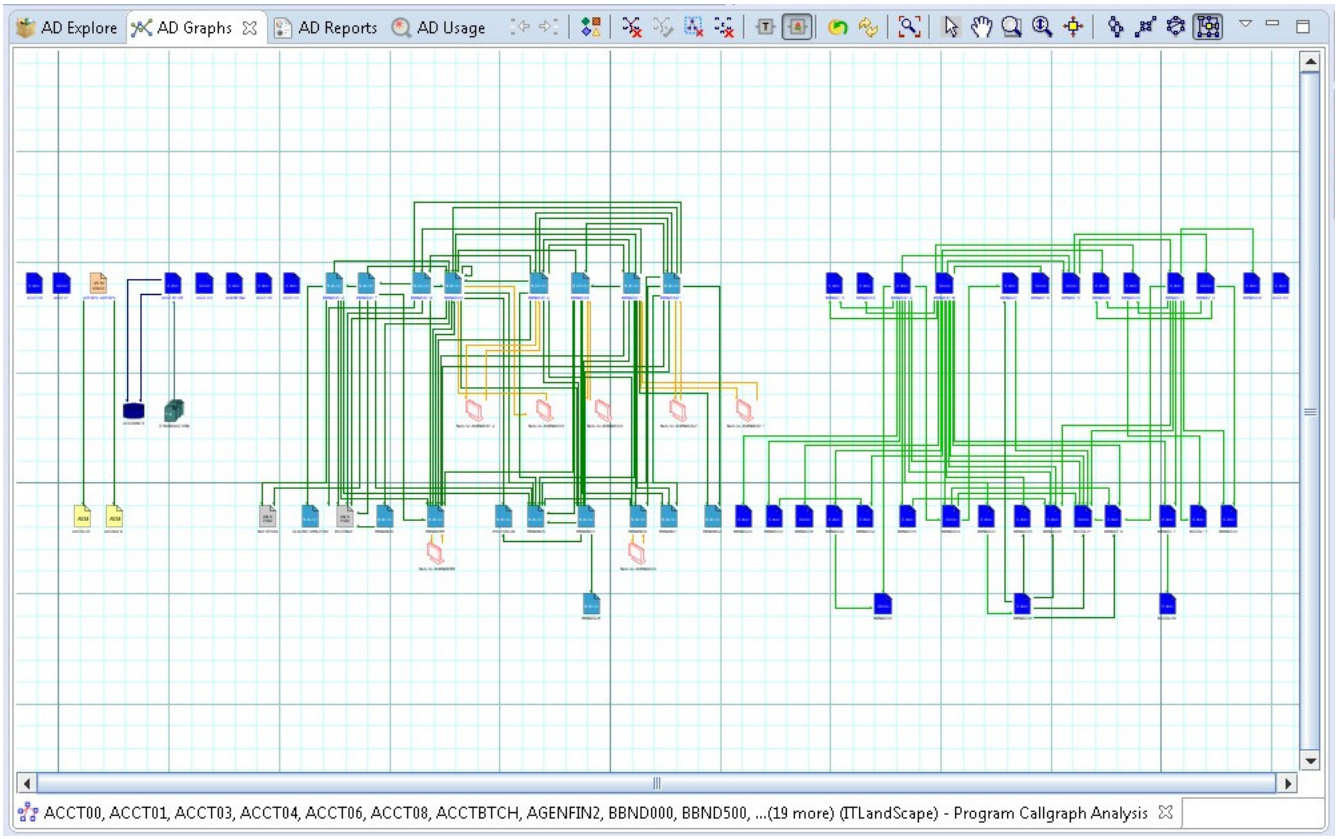


Figure 52. Orthogonal graph layout

Selecting graph elements

You can select multiple elements in the graph by dragging the mouse over the area to be selected. Different operations can then be performed on the selected objects such as printing, exporting.

Printing the graph

For maintenance purposes, the graph can be sent to a printer or to any other printing device connected to your computer.

To print the graph, from IBM AD Analyze client File menu, select **Print** or press "Ctrl+P".

The Print Manager window appears with the current graph shown. Click **Print Setup** in the upper left corner of **Print Manager** window to define the printing settings. Click **Print** in the upper left corner of **Print Manager** window to open the standard Windows Print window and print the graph.

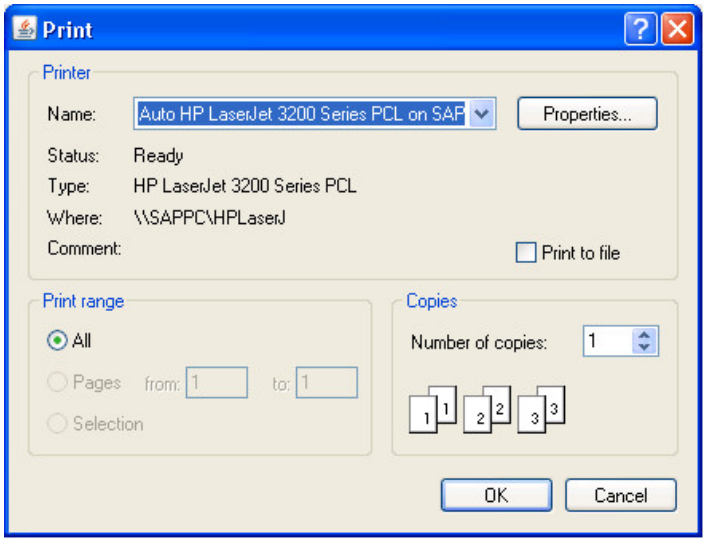


Figure 53. Print the graph

Exporting the graph

In addition to printing, the graph can also be exported to different file formats.

To export the graph, from **IBM AD Analyze** menu, select **Export Graph**. The **Export Graph** window appears.

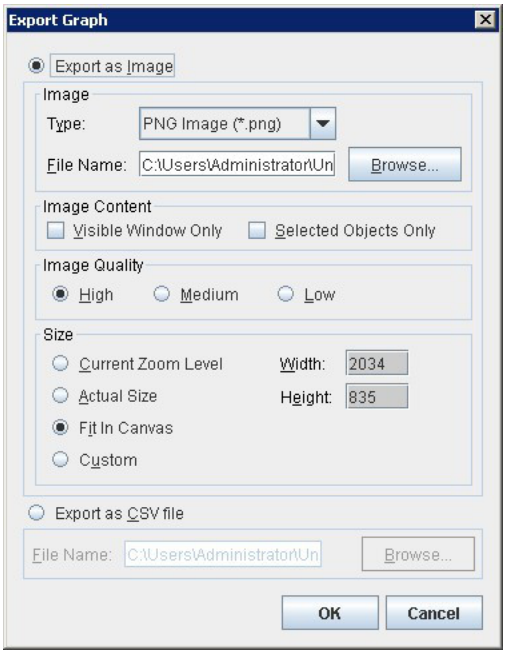


Figure 54. Export Graph

Image Area

In the **Type** field, select the type of file to be created. Supported types include *.jpg, *.pdf, or *.svg.

In the **File Name** field, the default file name and location is presented. The default file name is the name of the resource with the extension selected in the **Type** field. The default location is `C:\Program Files\eclipse`. Type in a new name for the file if applicable, and click **Browse** to select a new location for the file.

Image Content Area

Select one of the available options to define the image content.

Image Quality

Select one of the available options to define the quality of the exported image. A higher-quality image will, by definition, be larger in file size.

Size Area

Select one of the following to determine the size of the exported image:

- **Current Zoom Level**, the image is created at same zoom level as currently shown in AD Graphs window.
- **Actual Size**, the entire graph is printed at a zoom level of 100% regardless of the zoom level of the graph shown in AD Graphs window.
- **Fit in Canvas**, the image is created with the entire graph shown in one page of the file.
- **Custom**, the image is created in a size specified by the user. Enter the desired width and height for the image, in pixels, in the Width and Height fields.

Export as CSV file

Note: Export as CSV file option is only available for Program callgraphs generated with forward or backward options. It is NOT available for a Program callgraph generated for both call directions.

Select this option to export the graph as a CSV file. Use **Browse** to select the location where you want to export the graph.

Click **OK** to begin the export operation.

Program Flow

The Program Flow graph presents the internal structure for a program, including all referenced files, tables, screens, and other programs.

The following entities are visible depending on the language:

- Cobol – Sections, Paragraphs, PERFORM, and PERFORM THROUGH, Fall Through, GOTO, CALL, CICS XCTL, CICS LINK, CICS RETURN, CBLTDLI, CEETDLI.
- Natural – Main, Subroutines, PERFORM, CALL, CALLNAT.
- PL/I – Procs, CALL, GOTO, CICS XCTL, CICS LINK, CICS RETURN, PLITDLI.

To display the Program Flow graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a Program Flow graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click Program Flow to display the Program Flow wizard.

The Available Programs area displays the resources available for graph generation. To filter the available resources, click one of the buttons to the left of the list to display only certain types of resources. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists. Select a program from the available list then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. Paragraph names (COBOL) are shown as green rectangles to the left. Colored rectangles on the right represent external entities. The relationships between the paragraphs (for example, GoTo, Entry, Xctl) and external units (BMS map, programs) are denoted by color-coded edges or lines. For details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Program Flow graph toolbar

The **Program Flow** Graph toolbar has the following extra icons.

Table 16. Program flow graph toolbar

Button	Description
Flow Chart	Flow Chart is a toggle option. When this icon is selected, **IBM AD Analyze Client** displays the flow chart for the resource for which the flow graph is shown.
Change to Resources layout	After you tried out different layouts for the **Program Flow** graph (hierarchical, symmetric, circular, or orthogonal) use this button to go back to the initial resources layout.

Program Flow element pop-up menu

Right-click on an element from the Program Flow to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **Go to source**, **Show properties**, and **Filter**.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Screen Call Graph

The Screen call graph shows all programs that use a specific screen.

To display the Screen call graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the screen for which you want to generate a Screen call graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click Screen Call Graph to display the Screen Call Graph wizard.

The Available Screens area displays the resources available for graph generation. To filter the available resources, click one of the buttons to the left of the list to display only certain types of resources. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists then select the **Call direction**. By default, the graph is limited to three levels. Modify if applicable, then click **Finish**.

AD Graphs tab displays the generated graph. Input nodes are displayed in blue. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

Note: For large, complex graphs you can choose to restrict the number of programs displayed. You can specify a program beyond which subsequent programs must not be displayed. According to the type of database that is used for the projects, restricted nodes are displayed differently. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

By default, a limit icon marks the nodes for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all nodes that reach the level limit set in the wizard are marked with a limit icon regardless of whether further analysis results are available or not. The same criteria apply for displaying the limit icon in the Graph Inventory window.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory window presents a list of all the elements in the graph. Input nodes are displayed in blue. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Supported resource types for API calls

For the resource types that are supported for API calls, go to section Supported resource types for API calls in topic Program Callgraph.

Screen Call graph element pop-up menu

Right-click on an element from the Screen call graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Reports** and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Shared Resources Graph

This graph will be available after selecting two or more projects.

To display the Shared Resources graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the projects for which you want to generate a Shared Resources graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

After selecting two or more projects from the Explore Projects tab, the available graphs, reports, and usage analyses are displayed.

In the Shared Resources section, to the right of the Explore Projects view, double-click Shared Resources Graph to display the Shared Resources Graph wizard.

The Available applications area displays the applications available for graph generation in the selected projects. Use the arrow buttons to move the applications between the Available and Selected lists then click **Next** to display the second dialog window of the wizard.

The Available types area displays the types of resources available for graph generation. Use the arrow buttons to move the resource types between the **Available** and **Selected** lists then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Shared Resources graph element pop-up menu

Right-click on an element from the **Shared Resources Graph** to display the pop-up menu.

The options available in the pop-up menu depend on the type of element selected from the graph. Options available for all element types include **Show properties** and **Filter**. According to the selected element, **Expand** or **Collapse** options might also be available.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Transaction Call Graph

The Transaction call graph shows all programs that use a specific transaction.

To display the Transaction call graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a Transaction call graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click Transaction call graph to display the Transaction Call Graph wizard.

The Available Transactions area displays the resources available for graph generation. To filter the available resources, click one of the buttons to the left of the list to display only certain types of resources. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists then select the Call direction. By default, the graph is limited to three levels. Modify if applicable, then click **Finish**.

AD Graphs tab displays the generated graph. Input nodes are displayed in blue. To navigate the graph, use the buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

Note: For large, complex graphs you can choose to restrict the number of programs displayed. You can specify a program beyond which subsequent programs must not be displayed. According to the type of database that is used for the projects, restricted nodes are displayed differently. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

By default, a limit icon marks the nodes for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all nodes that reach the level limit set in the wizard are marked with a limit icon regardless of whether further analysis results are available or not. The same criteria apply for displaying the limit icon in the Graph Inventory window.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. Input nodes are displayed in blue. For details, see AD Graph Inventory View.

Note: In the **Graph Inventory** window, the names of the restricted nodes are displayed differently according to the database that is used for the projects. When working with graph database, restricted graph nodes are displayed in red and marked with a specific icon - a funnel. When working with SQL database, restricted nodes are displayed in red. Restrictions can be defined from **Preferences** window, for details on how to set these restrictions see Call graph restrictions.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Supported resource types for API calls

For the resource types that are supported for API calls, go to section Supported resource types for API calls in topic Program Callgraph.

Transaction Call graph element pop-up menu

Right-click on an element from the **Transaction Call Graph** to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **Show properties** and **Filter**.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

TWS Graph

The TWS Graph shows the scheduling graph starting from the selected job or jobs.

To display the TWS Graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the project that contains the resource for which you want to generate a TWS Graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Mainframe Graphs section, to the right of the Explore Projects view, double-click TWS Graph to display the TWS Graph wizard.

Use the options in the Input Options area to filter analysis inputs by either job name or application name. Alternatively, you can use the application Validity dates to filter the inputs for the TWS Graph. If no filter criteria is entered, the complete list of applications is presented. In the Graph Options area, use the available options to customize the appearance of the generated graph then press **Next** to display the second dialog window of the wizard.

The Available TWS Applications area displays the applications that are identified within TWS input resources. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the applications between the Available and Selected lists then click **Next** to display the third dialog of the wizard.

The Available TWS jobs area displays the TWS jobs available in the selected application. Use the **Name** field to filter the contents of a list. Use the arrow buttons to move jobs between the Available and Selected lists then click **Finish**.

To limit the number of applications to be displayed, click **Next** to display a fourth dialog of the wizard. Use the available options to set up the graph restrictions then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

TWS graph element pop-up menu

Right-click on an element from the TWS Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **Show properties** and **Filter**. According to the selected element, some or all of the following options might also be available: **Collapse ...**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Legacy Java Graphs

Cross Applications Call Graph

This graph is used to reveal interfaces and/or code flows between projects. Applications or projects are represented in the graph as nodes, while an edge between two applications represents one or more calls via a given call mechanism or interface type. The purpose of the edges is to show the various ways in which the applications communicate rather than how many calls per interface type. The number of calls represented by a given edge is shown to the user by a number next to the edge or a pop-up hint. The Cross Applications Call Graph can be generated for two projects (Java and mainframe).

To display the Cross Applications Call Graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the Java project which contains the resource for which you want to generate a Cross Applications Call Graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Cross Application Graphs section, to the right of the Explore Projects view, double-click Cross Applications Call Graph to display the Cross Applications Call Graph wizard.

A list of pre-defined Method calls is displayed. Use **Add** to add elements to the list if necessary then press **Next** to display the second dialog window of the wizard.

The Mainframe projects area displays the corresponding mainframe projects found on the server. Select one of the available mainframe projects then click **Finish** to generate the graph for the selected Java and mainframe projects.

AD Graphs tab displays the generated graph. To navigate the graph use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The Properties window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Cross Applications Call graph element pop-up menu

Right-click on an element from the Cross Applications Call Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Entity Diagram

The entity diagram presents a diagram for all the classes in the project that contain an @Entity annotation. Each graphic element shown in the Entity Diagram represents a class or a type of relationship

within the selected entity. Various operations can be performed on the elements in the Entity Diagram. Some operations are common to all types of elements in all graphs; others only apply to certain types of elements.

To display the Entity Diagram, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the Java project for which you want to generate an Entity Diagram. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Java Graphs section, to the right of the Explore Projects view, double-click Entity Diagram. The Entity diagram is presented in **AD Graphs** tab. To navigate the graph, use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Entity Diagram element pop-up menu

Right-click on an element from the Entity Diagram to display the pop-up menu.

The options available in the pop-up menu depend on the type of element selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Usage** analyses.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Imports Analysis Graph

The Imports Analysis Graph presents the inheritance relationships (imported, inherited, implemented) among all the entities in the application analyzed. Each graphic element that is shown in the Imports Analysis Graph represents a type of resource or a type of relationship.

To display the Imports Analysis Graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the Java project for which you want to generate an Imports Analysis Graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Java Graphs section, to the right of the Explore Projects view, double-click Imports Analysis graph to display the Imports Analysis graph wizard. By default, all components (imported, inherited, and implemented) are included in the graph. Deselect the components that you want to exclude from the analysis then press **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Imports Analysis graph element pop-up menu

Right-click on an element from the Imports Analysis Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Graphs**.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Legacy Java Call Graph

The Java Call Graph presents the chain of possible calls at runtime in Java. The nodes in the graph represent the project components, while the edges (arrows) represent relationships between the components. Both nodes and edges are color coded to show the type of component or relationship.

To display the Java Call Graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the Java project for which you want to generate a Java call graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Java Graphs section, to the right of the Explore Projects view, double-click Java call graph to display the Java call graph wizard.

The Available resources area displays the resources available for graph generation. To filter the available resources click one of the buttons to the left of the list to display only certain types of resources. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory pane presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element that is selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Java Call graph element pop-up menu

Right-click on an element from the Java call graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Graphs**.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Resource Dependencies Graph

The Resource Dependencies Graph shows the links between the special entities present in the special web category (objects such as *.jsp, *.js, *.css, *.html files). This graph is only available for JSP and HTML object files. For the Resource Dependencies Graph, one additional operation is available: Java Call Graph.

To display the Resource Dependencies Graph, in Application Discovery Browser perspective, from the Projects list in the central part of the Explore Projects view, select the Java project for which you want to generate a Resource Dependencies Graph. For details on perspectives and their components, see IBM AD Analyze Client Reference.

In the Java Graphs section, to the right of the Explore Projects view, double-click Resource Dependencies Graph to display the Resource Dependencies Graph wizard.

The Available resources area displays the resources available for graph generation. To filter the available resources click one of the buttons to the left of the list to display only certain types of resources. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish**.

AD Graphs tab displays the generated graph. To navigate the graph, use buttons in the graph toolbar. For details, see Graphical Analysis Toolbar. To show details on the meaning of the elements of the graph, use the **Legend** button. For details on the graph legend, see Showing the Graph Legend.

AD Graph Overview window presents an overview of the entire graph, which can be useful when you work with large graphs. For details, see AD Graph Overview Window.

AD Graph Inventory window presents a list of all the elements in the graph. For details, see AD Graph Inventory View.

The **Properties** window displays the properties of the element selected from the graph. For details, see Properties View.

Use **Export Graph** and **Print** (from the Eclipse toolbar) to export the graph to a portable format or print it. For details, see the sections Exporting the Graph and Printing the Graph.

Resource Dependencies graph element pop-up menu

Right-click on an element from the Resource Dependencies Graph to display the pop-up menu.

The options available in the pop-up menu depend on the type of element that is selected from the graph. Options available for all element types include **View analysis source**, **Show properties**, and **Filter**.

According to the selected element, some or all of the following options might also be available: **Annotations**, **Reports**, and **Graphs**.

The **Filter** options in the pop-up menu allow the user to toggle on/off different elements of the graph.

For details on the options available for all element types, see Graph element pop-up menu.

Reports

The **Report** component will generate the reports specified in the configuration file.

Mainframe Reports

Generating a Report

In Explore Projects view, go to Mainframe Reports list in the pane to the right. In the report list, double-click the wanted report. The report wizard is displayed presenting the resources available in the current project.

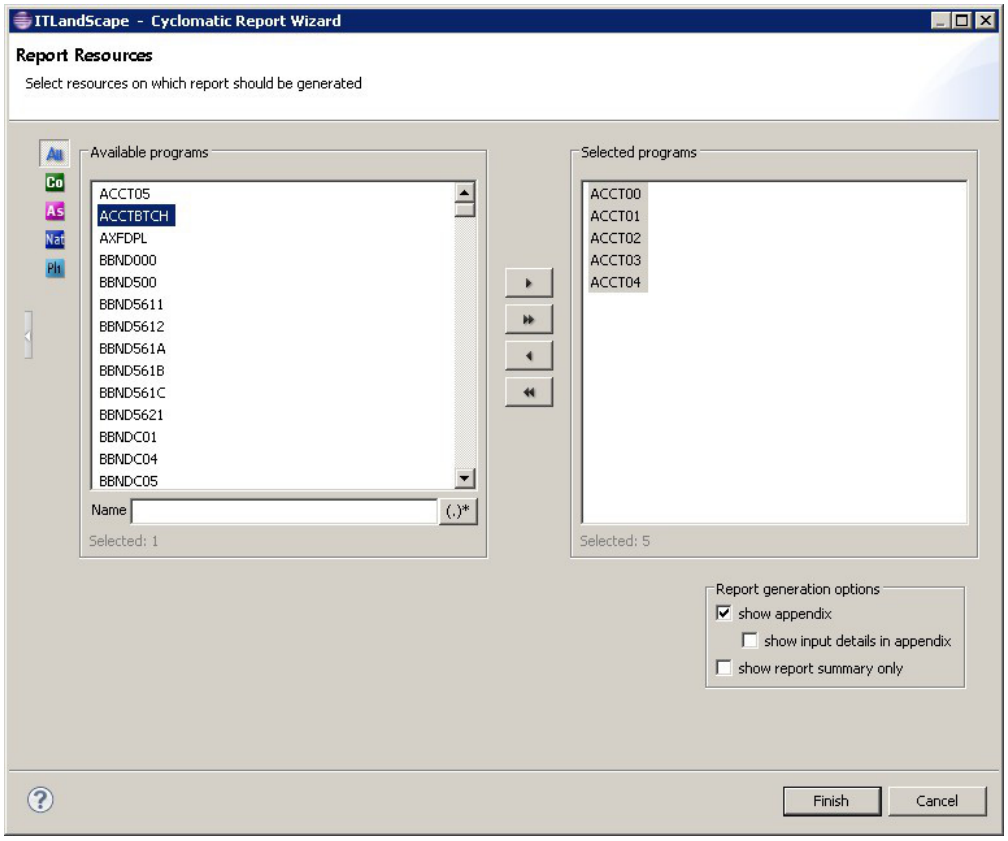


Figure 55. Cyclomatic Report Example

The Available area displays the resources available for report generation and allows the user to select the resources to be included in the report. The resources from the list can be filtered, by using the **Name** field, in the following two ways:

- Writing the name of the resource.
- Using regular expressions.

Examples of regular expressions that can be used for a given string `CICS9.SAMG1.DFHAIPL1`:

- `^.*\.DFHAIPL1` that is equivalent to `^.*DFHAIPL1`
- `CICS9\.SAM.*`
- `CICS9\.SAMG1\.DF.*`
- `^.*\.DFHA.*`
- `^.*\.(?i)df.*`

Important: The filtering that uses regular expressions is case-sensitive. For case-insensitive match, the `(?i)` token can be used.

Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the report appendix is displayed by default. To include input details in the appendix, select the option show input details in appendix. To show only the report summary, select the appropriate option.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

Note: For certain reports, extra dialog windows are available allowing the selection of target sources for the report.

The Generating Report message appears, showing the progress of the report generation. For a large volume of data, the report generation may take some time. Therefore, **IBM AD Analyze Client** provides an option of running the report in the background so that you can continue working while the report is generated. To run the report in the background, click **Run in Background** in the **Generating Report** dialog.

While the report is being generated, icons in the Status bar indicate the type of report that is being generated and the progress of the operation. Click this icon to see the progress in the **Progress** tab.

After the report was generated, it is presented under a tab that is named with the report name in **AD Reports** tab.

Report Toolbar

The toolbar available for the reports is displayed as in the following image.

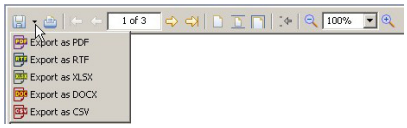


Figure 56. Report toolbar

It allows the user to perform the following operations on the generated report:

Table 17. Report toolbar buttons

Button	Description
Export	Export the report in PDF, RTF, XLSX, DOCX, or CSV format.
Print	Print the report on a printer that is connected to your system.
Fist page	Jump to the first page of the report.
Previous	Go to the previous page of the report.
	Number of the current page.
Next	Go to the next page of the report.
End	Jump to the last page of the report.
Actual size	Present the report in standard size.
Fit page	Present the report to fit to page.
Fit width	Present the report as to fit to width.
Go back	Return to the previous location in the report.
Zoom out	Zoom out (in increments of 25).
	Current zoom percentage. Select from the drop-down list to zoom to 25% - 200% (in increments of 25).
	Zoom in (in increments of 25).

Complexity Reports

Cyclomatic Complexity Report
The Cyclomatic complexity Report (or the McCabe Complexity Report) presents Cyclomatic complexity (a broad measure of soundness and confidence for a program) for the selected project entity. Cyclomatic complexity measures the number of linearly-independent paths through a program module.

To define the complexity thresholds for the Cyclomatic Report for the current project, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Then, select **Cyclomatic Reports** and right-click on the **Cyclomatic Report** and select **Show settings**. In the Report settings dialog window, the **Thresholds** tab presents the complexity thresholds (Easy, Moderate, and Difficult) to be considered when the application calculates the cyclomatic complexity for the current analysis. Modify the default values as needed. The threshold for Difficult cannot be modified. A color is assigned by default to each complexity level; it is the color that is used in the report diagrams to represent a particular complexity level. To change the default color, double-click the color chip and then select the color you want. Use the **Restore defaults** button to return to the default settings.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Complexity Reports then double-click the wanted report to display the report wizard. The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the selection indicates that report appendix is displayed by default. To include input details in the appendix, select the option show input details in appendix. To show only the report summary, select the appropriate option.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

In Complexity Score Summary, this report presents two charts: the Complexity percentage distribution chart and the File complexity distribution chart. For **COBOL, PL1, Natural** it is computed as follows:

```
Incrementing by 1 the total number of statements that contain flow decisions
```

Halstead Complexity Report
The Halstead Complexity Report generates software metrics that are computed statically without program execution. Halstead complexity can be used to compare the complexity between two programs or two applications.

To define the Halstead Complexity Report settings for the current project, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Then, select Complexity Reports, right-click on the Halstead Report and from the pop-up menu select **Show settings**.

In the Report settings dialog window, the **Operators** tab shows if a statement type is considered operator or operand when **IBM AD Analyze** calculates the Halstead complexity. By default, all statement types are considered operators. Adjust the selections according to specific needs of your project. Use **Check all** and **Clear all** to manage the selection and **Restore defaults** to return to the default selection.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Complexity Reports then double-click **Halstead Complexity Report** to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data you want to include in the appendix: **show input details in appendix**, **show operators in appendix**. To show only the report summary, select the appropriate option.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents charts that show the Program length distribution, the Vocabulary size distribution, the Program volume distribution, the Difficulty level distribution, the Program level distribution, the Effort to implement distribution, the Time to implement distribution, and the Number of bugs distribution per program.

The Halstead Complexity Report details section shows for each program a chart that presents Program length, Vocabulary size, Program volume, Difficulty level, Program level, Effort to implement, Time to implement, and Number delivered bugs.

The following formulas are used to measure different indicators:

- **Program length** - measures the total number of operator occurrences and total number of operand occurrences, representing the length metric for the program.

```
N = N1 + N2
```

- **Vocabulary size** - measures the total number of unique operators and operands within the program.

```
n = n1 + n2
```

- **Program volume** - measures the minimum number of bits required for coding the program.

```
V = N * log2(n)
```

- **Difficulty level** - measures the level of difficulty in the program.

```
D = (n1 / 2) * (N2 / n2)
```

- **Program level** - measures the level at which the program can be understood.

```
L = 1 / D
```

- **Effort** - measures the mental effort that is required to develop or maintain the program. The lower the value of this measure, the simpler the program would be to change.

```
E = V * D
```

- **Time to implement** - measures the amount of time to implement the algorithm.

```
T = E / 18
```

- **Number of delivered bugs** - measures the estimated number of errors in the program.

```
B = ( E exp (2/3) ) / 3000
```

Where:

- **n1** = the number of distinct operators
- **n2** = the number of distinct operands
- **N1** = the total number of operators
- **N2** = the total number of operands

Heuristic Complexity Report
The Heuristic complexity Report determines the complexity of an application by assigning values to each type of statement available in the programming language. In the setup screen, the statements are arranged in groups to allow giving a value to all statements within the group.

Once the values are determined, the thresholds must be set to define the boundaries between simple, moderate, and complex programs. At complexity measurement time, each program complexity is determined by multiplying each statement weight by the statement occurrence number and then

summing up all values for that program. The final complexity value is then compared to the thresholds to determine the degree of complexity (simple, moderate, or complex).

Heuristic complexity can be used to determine the effort of a specific software change by focusing on specific statements. For example, giving a weight of zero for all statements except SQL statements helps to measure the complexity of a change that relates only to SQL statements within your application. The weight can represent for example man/hours, providing a time estimation.

To define the Heuristic Complexity report settings for the current analysis, in Explore Projects view, go to the Mainframe Reports list in the pane to the right, then select Complexity Reports. Right-click the Heuristic Report and select **Show settings**. The Heuristic Report settings dialog window presents the weight and complexity (Simple, Moderate, and Complex) of each statement type. It contains two tabs: **Thresholds** and **Weights**.

The **Thresholds** tab presents the complexity thresholds (Simple, Moderate, and Complex) to be considered when IBM AD Analyze calculates the Heuristic complexity for the current analysis. Modify the default values as needed. The threshold for Complex cannot be modified. A color is assigned by default to each complexity level. It is the color that is used in the report diagrams to represent a particular complexity level. To change the default color, double-click the color chip and then select the desired color. Use the **Restore defaults** button to return to the default selection.

The **Weights** tab presents the weight of each statement type. By default, all statement types are assigned the value "1". You can change the weight of a single statement or of a group of statements.

Use **Restore defaults** button to return to the default selection.

To generate the Heuristic Complexity report, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Select Complexity Reports then double-click the desired report to display the report wizard. The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**, **show weights in appendix**. To show only the report summary, select the appropriate option.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Complexity percentage distribution and the File complexity distribution.

The Report details section shows for each program the number of statements, the Program weight, and the complexity level attributed (according to the values set in the Heuristic Report Settings dialog).

Maintainability Index Report

The Maintainability Index Report is a measurement that is intended to track maintainability and to indicate when it becomes cheaper or less risky to rewrite code instead of changing it. The Maintainability Index report presents the combined maintainability, McCabe, and Halstead complexity measurements for the current project.

To define the Maintainability Index report settings for the current project, in Explore Projects view, go to the Mainframe Reports list in the pane to the right then select Complexity Reports. Right-click the Maintainability Index Report and select **Show settings**.

In the Report settings dialog window, the **Thresholds** tab presents the maintainability thresholds (Difficult, Moderate, and Easy) to be considered when IBM AD Analyze calculates the Maintainability Index for the current analysis. Modify the default values as needed. The threshold for "Easy" cannot be modified.

A color is assigned by default to each maintainability level; it is the color that is used in the report diagrams to represent a particular maintainability level. To change the default color, double-click the color chip and then select the desired color. Use **Restore defaults** button to return to the default selection.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Complexity Reports then double-click the desired report to display the report wizard. The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. To show only the report summary, select the appropriate option.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Maintainability index percentage distribution and the File maintainability index distribution.

The Report details section shows for each program the computed Maintainability index value and the maintainability level attributed (according to the values set in the Maintainability Index Report Settings dialog).

The Maintainability Index Report is computed by using the following formula:

```
171 - 5.2 * ln(Halstead Volume) - 0.23 * (Cyclomatic Complexity) - 16.2 * ln(Number of
statements)
```

Where:

- **Halstead Volume** is computed as:

  ```
  V = N * log2(n)
  ```

  Where:

  – **N** represents the **Program Length** and is computed as:

  ```
  N = N1 + N2
  ```

  N1 = the total number of operators

  N2 = the total number of operands

  – **n** represents the **Vocabulary size** and is computed as:

  ```
  n = n1 + n2
  ```

  n1 = the number of distinct operators

  n2 = the number of distinct operands

- **Cyclomatic Complexity** is computed as:

  ```
  Incrementing by 1 the total number of statements that contain flow decisions
  ```

Datasets Reports

Data Flow Through Datasets Report
The Data Flow Through Datasets Report shows the data flow within a batch process, starting with a physical dataset that can be either at the start of the process (forward data flow) or at the end of the data flow (backward data flow) or in the middle of it (backward and forward data flow).

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Datasets Reports then double-click the desired report to display the report wizard. The Available Datasets area lists the datasets available in the selected project. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the datasets between the Available and Selected lists.

In Report generation options area, if you select the option **generate detailed info**, the report presents all identified paths between two datasets. If you do not select this option, the report presents only one path between two datasets. The report appendix is displayed by default. If you select **show input details in appendix**, a list of all datasets that were used to generate the report is included in the appendix.

In Limit data flow area, by default, a complete report is generated. If you want to limit the analysis to X depth, check the option **Threshold level** and enter the desired value in the corresponding field. From the Data flow direction area, select one of the available options **Forward**, **Backward**, or **Both**.

Click **Next** to display the second dialog window of the wizard.

Select the datasets that you want to exclude from the report. Click **Finish** to generate the report.

Excluded nodes are marked with a "!" sign.

By default, an asterisk sign, "*", marks the chains for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all chains that reach the level limit set in the wizard are marked with an asterisk regardless of whether further analysis results are available or not. The meaning of the asterisk sign "*" in the context of each report is explained in a note at the end of the report.

Datasets Record Structure Report

The Datasets Record Structure report presents the record that defines the file structure and also where it is being used in the programs.

To generate this report, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Select Datasets Reports then double-click the desired report to display the report wizard.

The Available datasets area displays the datasets available for report generation. Use the **Name** field to filter the contents of the list.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**.

Use the arrow buttons to move the datasets between the Available and Selected lists then click **Finish** to generate the report.

Dead Code Reports

Unreferenced Programs

This report lists all programs that are not referenced by other programs within the project. These programs must be tested to determine whether they are main programs (and therefore are not referenced) or whether they are not used and can be considered dead code.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Dead Code Reports then double-click the desired report to display the report wizard. The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data must be included in the appendix: **show input details in appendix**.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents a chart that shows the Programs usage ratio.

The Report details section shows the list of unreferenced programs that were found.

Unused Routines Within Programs

This report lists all routines within a program that are not referenced within the program. These routines cannot be reached at runtime and therefore are considered dead code.

To generate this report, in Explore Projects view, go to Mainframe Reports list in the pane, to the right. Select **Dead Code Reports** then double-click the desired report to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data must be included in the appendix: **show input details in appendix**. The option **Filter out unused EXIT routines for COBOL programs** is selected by default; deselected it if you want to include these routines in the report. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show usage ratio of programs that have unused routines, and the unused routines occurrences in program.

The Report details section shows for each program: the routine name, the number of statements, and the line in the program where the routine is located.

Impact Reports

Field Expansion Report

This report presents the relationships among the resources of an application so that the impact of a field expansion operation can be understood.

To generate this report, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Select Impact Reports then double-click the desired report to display the report wizard.

In the first dialog window of the wizard, specify the location of the variable for which you want to generate the Field Expansion report or browse for a table field.

Note: If you choose to start the impact report for an SQL table field, the report is generated on a variable that is mapped with the table field and not on the table field.

If you want to limit the analysis to X depth, check the option **limit expansion depth** and enter the desired value in the corresponding field then click **Next**.

In the second dialog window of the wizard, select the type of resource that contains the variable - program or include - then click **Next**.

According to the selections made in the previous dialog, a list of programs or includes is displayed. Use the field to filter the list and locate the desired resource. Once you selected the desired resource, the **Next** button becomes available. Click it to display the next dialog window of the report wizard.

A list of variables that were found in the selected resource is presented. Use the field to filter the list and locate the desired variable, then select it and click **Next** to display the next dialog window of the report wizard.

Note: In case you selected the **include** type of resource, only the variables from the first program are listed.

Use this dialog to specify variables that must be EXCLUDED from the report. If you do not want to exclude variables from the report, click **Finish**. If you want to exclude certain variables from the report, follow the instructions below.

Use **Import** to import a file that contains the variables that you want to exclude from the report. The variable list must be saved in a *.txt file, one variable on each line.

Enter the name of the variable in the available field then press the **Add** button to add it to the list.

Alternatively, use **Browse** to locate the variable. Click **Browse** to select the type of resource that contains the variable and display the next dialog window of the report wizard.

Select the type of resource - program or include - then press **Next** to display the next dialog window of the report wizard. According to the selection made in the previous dialog window, a list of programs or includes is presented. Select a program or an include then click **Next** to display the next dialog window of the report wizard.

A list of variables that were found in the selected resource is presented. Select one of the variables from the list then click **Add** to add it to the list of variables to be excluded from the report.

The variables that were selected for EXCLUSION are displayed in the central part of the dialog window. For variables added via **Browse** and **Import** buttons, the name of the resource (program or include) containing the variable is displayed as well as the variable's position (line ...) within the resource.

Repeat the procedure that is presented above to add to the list as many variables as you need. To remove a variable from the list, select it then click **Delete** button. Click **Finish** to generate the Field Expansion Report.

After **Field Expansion Report** is generated, in the **Report summary** section, of the **Table of contents**, different entries are displayed. The entries can be: Variables, Programs, SQL Table, File, and Sources.

For all entries, there are dedicated sections that contain their summaries. For programs, there is a dedicated section that is called **Programs summary page** and which contains the name of the programs that are impacted. Some of the programs that are displayed are underlined and some are not.

The programs that are not **underlined** are the ones that represent the **starting point** for the selected variable.

The **underlined** programs are the ones that are the **final point** for the selected variable. If an underlined program is selected, you are redirected to the **Report Details** section where the impact analysis of the program is displayed.

Best Practices on how to use the Field Expansion Report

Introduction

Field Expansion Report presents the relationships among the resources of an application so that the impact of a field expansion operation can be understood. Field Expansion tracks the format of the data through all possible statements to understand the possible impact of a format change.

Purpose

The goal is to understand the impact of modifying the format of data that might flow through programs by limiting the expansion depth and to reiterate the process at every cycle. After the first cycle, it is recommended to create an exclude list for the items that are not related to the starting variable.

Running with no depth might be time consuming. It takes a long period of time to generate the report and it is difficult to read the information present in all pages.

Basic Principles

As a best practice, it is highly recommended to start running the report with a small depth, for example you can use 5 as a value.

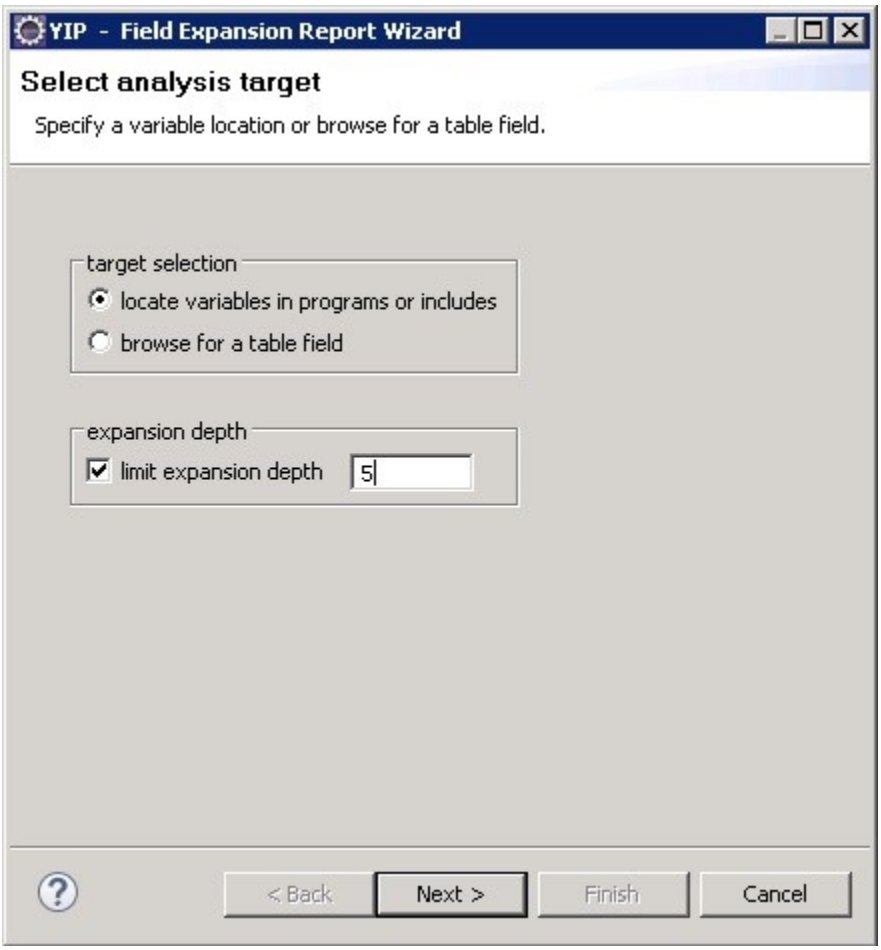


Figure 57. Select Analysis Target Window

Once it is generated, start to analyze the report and it is recommended to create an exclude list for the items that are not related to the starting variable (e.g. written into the same message or log record).

Note:

- The exclude list prevents the report from expanding to other content domain and inflate the report with unwanted results.
- It is highly recommended to create the exclude list as a simple text file and have it loaded whenever the report on the same inputs is executed.

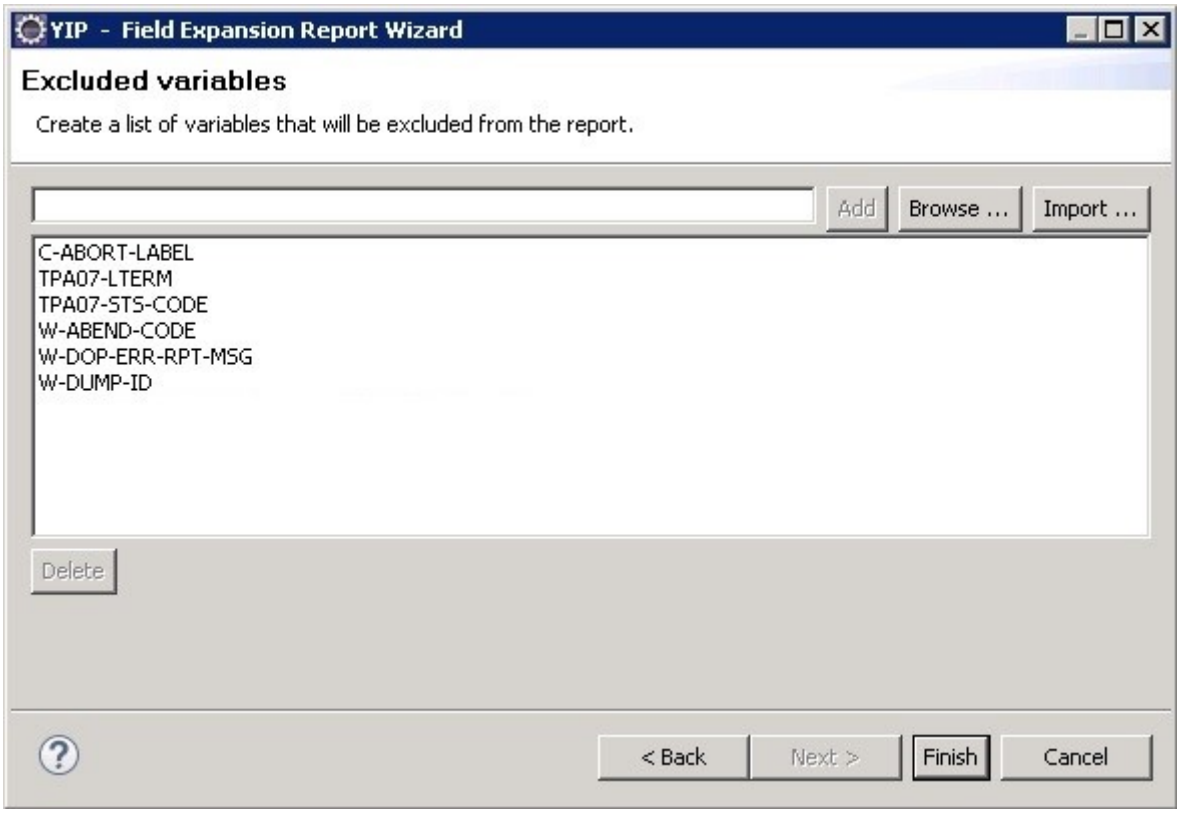


Figure 58. Excluded Variables Window

Run the report again by increasing the depth to 10 and by loading the exclude list.

Analyze the report again and if needed update the exclude list with the new items that are not related to the starting variable.

Perform these steps until you obtain a simplified report that can provide the necessary information.

Field Usage Report

This report presents the relationships among the resources of an application so that the impact of renaming a field can be understood.

To generate this report, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Select Impact Reports then double-click the desired report to display the report wizard.

In the first dialog window of the wizard, select the target of the analysis - either locate variables in programs or includes, or browse for a table field - then click **Next** to display the second dialog window of the wizard.

Note: If you choose to start the impact report for an SQL table field, the report is generated on a variable that is mapped with the table field and not on the table field.

Choose the type of source file - program or include - for variable selection then click **Next**.

Note: In case you selected the **include** type of resource, a list of include items is presented. Select a single entry from the available list, then press **Next**. A list of programs that uses the selected include resource is presented. Use CTRL to select non-consecutive programs from the list.

Regardless of the selections made in the previous dialogs, a list of programs is displayed. Use the field to filter the list and locate the desired resource. Select the resource, then click **Next** to display the next dialog window of the report wizard.

A list of variables that were found in the selected resource is presented. Use CTRL to select non-consecutive variables from the list. Use the field to filter the list elements and to locate the desired variable then select it and click **Finish** to generate the report.

Note: In case you selected the **include** type of resource, only the variables from the first selected program are listed.

Impact Analysis Report

The Impact Analysis report allows the developer to understand the consequences of modifying a certain resource. The impact analysis determines who uses the objects and presents the dependencies between the objects.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Impact Reports then double-click the desired report to display the report wizard.

In the first dialog window of the report wizard, specify the location of the variable for which you want to generate the Impact Analysis report, or browse for a table field.

Note: If you choose to start the impact report for an SQL table field, the report is generated on a variable that is mapped with the table field and not on the table field.

Then, select the impact direction (omnidirectional, forward, or backward). If you want to limit the analysis to X depth, check the option **limit expansion depth** and enter the desired value in the corresponding field then press **Next**.

Select the type of resource - program or include - that contains the variable for which you want to perform the analysis, then press **Next**.

Note: In case you selected the **include** type of resource, a list of include items is presented. Select a single entry from the available list, then press **Next**. A list of programs that uses the selected include resource is presented. Use CTRL to select non-consecutive programs from the list.

Regardless of the selections made in the previous dialogs, a list of programs or tables is displayed. Use the field to filter the list and to locate the desired resource. Select the resource then click **Next** to display the next dialog window of the wizard.

A list of variables that were found in the selected resource is presented. Use CTRL to select non-consecutive variables from the list. Use the field to filter the list and locate the desired variable then select it and click **Next** to display the next dialog window of the report.

Note: In case you selected the **include** type of resource, only the variables from the first selected program are listed.

Use this dialog to specify variables that must be EXCLUDED from the report. If you do not want to exclude variables from the report, click **Next**. If you want to exclude certain variables from the report, follow the instructions below.

Enter the name of the variable in the available field then press **Add** to add it to the list.

Use **Import** to import a file that contains the variables to be excluded from the report. The variable list has to be saved in a *.txt file, one variable on each line.

Alternatively, use **Browse** to locate the variable. Click the **Browse** button to select the type of resource that contains the variable and to display the next dialog window of the report wizard.

Select the type of resource - program or include - then press **Next**.

According to the selections made in the previous dialog, a list of programs or includes is displayed. Select the resource then click **Next**.

A list of variables that were found in the selected resource is presented. Select one of the variables from the list then click **Add** to add it to the list of variables to be excluded from the report. The variables that were selected for EXCLUSION are displayed in the central part of the dialog window. For variables added via **Browse** and **Import** the name of the resource (program or include) containing the variable is displayed as well as the variable's position (line ...) within the resource.

Repeat the procedure that is presented above to add to the list as many variables as you need. To remove a variable from the list, select it then use **Delete**. Click **Next** to display the next dialog window of the report wizard.

Use this dialog to specify which datasets must be excluded from the analysis. Use the arrow buttons to move the datasets between the Available and Selected lists then click **Finish** to generate the Impact Analysis report.

After **Impact Analysis Report** is generated, in the **Report summary** section, of the **Table of contents**, different entries are displayed. The entries can be: Variables, Programs, SQL Table, File, DAL, Dataset, Impacted jobs, Impacted AAuto Networks, String Literal, Initial Value, Figurative Constant, and Sources.

For all entries, there are dedicated sections that contain their summaries. For programs, there is a dedicated section that is called **Programs summary page** and which contains the name of the programs that are impacted. Some of the programs that are displayed are underlined and some are not.

The programs that are not **underlined** are the ones that represent the **starting point** for the selected variable.

The **underlined** programs are the ones that are the **final point** for the selected variable. If an underlined program is selected, you are redirected to the **Report Details** section where the impact analysis of the program is displayed.

The Impact Analysis report takes into consideration the COBOL Reference Modification to move partial data from source field to target field. In the reference modification, the first position indicates the starting position and the second position indicates the length of the variable.

Impact Analysis Report for PL/I

The Impact Analysis report allows the developer to understand the consequences of modifying a certain resource. The impact analysis for PL/I shows the dataflow from the starting point to the end point, in the selected (forward or backward) direction. This report uses data stored in graph database. The following relationships are shown in the report:

- statements that actually transfer data (like assignments);
- statements that read from or write to external resources (like tables and files that are mapped further with datasets);
- parameter passing across internal procedures;
- parameter passing across external procedures;

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Impact Reports then double-click the desired report to display the report wizard.

In the first dialog window of the report wizard, specify the location of the variable for which you want to generate the Impact Analysis report, or browse for a datasource type: dataset or SQL table. If you select SQL table an additional wizard dialog is available where you can select a SQL field.

Note: If you choose to start the impact report for an SQL table field, the report is generated on a variable that is mapped with the table field and not on the table field.

Then, select the impact direction (forward or backward). If you want to limit the analysis to X depth, check the option **limit expansion depth** and enter the desired value in the corresponding field then press **Next**.

Select the type of resource - program or include - that contains the variable for which you want to perform the analysis, then press **Next**.

According to the selections made in the previous dialog, a list of programs, includes, SQL tables or datasets is displayed. Use the field to filter the list and to locate the desired resource. Select the resource then click **Next** to display the next dialog window of the wizard.

A list of variables that were found in the selected resource is presented. Use CTRL to select non-consecutive variables from the list. Use the field to filter the list and locate the desired variable then select it and click **Next** to display the next dialog window of the report.

Use this dialog to specify resources that must be EXCLUDED from the report. The list of entities available for exclusion presents variables, datasets, SQL tables, and SQL fields.

The exclude feature for Impact analysis report is done by adding a file with the details of the resources we want to exclude from report. The file structure is: `resource_type,resource_name,`

[resource_detail_type,resource_detail]. Each line keeps information only for a resource. Each type of resource has a specific list of pairs:

- for variables: variable, variable_name, [program, program_name, forefather, forefather_name];
- for datasets: dataset, dataset_name;
- for SQL tables: sqltable, sqltable_name;
- for SQL fields: sqlfield, sqlfield_name, [sqltable, sqltable_name].

Note: Here is an example of a file containing details of the resources that mus be excluded from the report:

```
variable, CUSTID-REC
variable, CUSTID-REC, program, ACCT03, forefather, MISC
variable, CUSTID-REC, forefather, MISC
sqltable, ACCOUNTS
sqlfield, SCITY
sqlfield, SCITY, sqltable, CUSTOMER
```

The results of the report depend on the type of statement selected for analysis.

The Report summary section includes a Variables summary page, a Programs summary page, a File summary page, a Dataset summary page, a SQL Field summary page, and a Sources summary page. These pages present the input (selected variables or programs) for the report and the impacted resources (variables, files, datasets, SQL fields, and programs).

The Report details section presents the input variable and the final datasource for the variable together with all the steps in between.

Example 1: statements that actually transfer data (assignments). We want to show the impact of modifying the variable LINE. An Impact analysis of variable LINE presents the data flow that reaches to the variable LINE, through the assignment statements. The Report summary section presents information about entities that are impacted, that is, the list of sources being touched. In the Report details section, the definition source and the program that started the flow are presented. If we click on PL1: Assignment statement, we will see the statement highlighted in the source code in the Editor view.

Example 2: statements that read or write to external resources (like SQL table fields and Files). This level allows the user to get an overview of how data flows through the system at a level that is closer to the business view. This time we want to show the impact of modifying a SQL Table Field. An Impact analysis of SQL field presents the flow of data beginning with the starting point and ending with the SQL table field name - that is suffixed with the SQL table name. After the report is generated, clicking on the SQL insert, we will see the INSERT INTO statement (that is used to insert new records in a table x, the fields names and the values to be inserted).

In case you select statements that read from or write to files that are mapped with datasets: an Impact analysis of a dataset presents the dataflow beginning with the starting point and ending with the file. The file that is written has a DDCard association. Clicking on the DDCard association we'll see JCL Data definition (DD) statement that defines the datasets.

Example 4: Parameter passing across internal procedures and external procedures, the following data is shown:

- the program that started the flow;
- table with columns: "from", "to" "impact type" describing the flow of data beginning with the starting point and ending with the program;
- subsection with possible chains of calls and data flows between programs that start or get to the current program;
- subsection with list of variables impacted in the program;
- subsection with list of programs impacted and how variables in the current program map to defined parameters of the called program.

Inventory Reports

Application Inventory Report

This report presents the list of resources that were found in the selected application or applications(s).

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Inventory Reports then double-click the desired report to display the report wizard.

A list of defined applications is displayed. Select the applications for which you want to generate the report.

In Report generation options area, the selection indicates that the report appendix is displayed by default. To show only the report summary, select the appropriate option. Click **Finish** to generate the report.

The Report summary section presents two charts that show Totals and Averages, including number of routines, number of resources used (files, tables, screens), number of includes, number of statements, number of calls, and usage details for each resource according to the different access statements. All information is grouped by application.

The Report details section shows the list of components that are associated to each application.

Members Inventory Report

This report presents a list of all the members of the current project.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Inventory Reports then double-click the desired report to display the report wizard.

The Available MF Libraries area displays the MF libraries that are available for report generation. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the arrow buttons to move the MF libraries between the Available and Selected lists then click **Finish** to generate the report.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. To show only the report summary, select the appropriate option.

The Report summary section shows the following information:

- The total number of members and the total number of lines.
- For each member type, the summary displays the total number of members per type and the total number of lines.
- Total number of members per library and total number of lines.
- For each member type with the specific library, the summary displays the total number of members per type and total number of lines.

The Report details section of the report contains a list of all members with information such as Member name, type, no. of lines, and so on.

Missing Components Report

This report presents a list of all missing components for the selected programs such as called programs or SCL procedure that are missing from the current project.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Inventory Reports then double-click the desired report to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button

to the left of the dialog window. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**.

The Report details section of the report contains two lists:

- By program – showing for all selected programs, all the missing components.
- By component – showing for all components, all the programs that call those components.

At the end of the report, in the Report summary section, the total number of missing components is presented.

Programs Inventory Report

The **Programs Inventory** report presents a detailed list of all the programs in the current project as well as details about each program.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select **Inventory Reports** then double-click the desired report to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. To show only the report summary, select the appropriate option.

The Report summary section presents Totals and Averages.

The Report details section shows for each program, the total number of **Backfired Function Points**, total number of calls, total number of statements, and total number of includes. Also, a list of all the includes identified is displayed.

In order to calculate the values, the following formula is used: number of statements / k, where k is the constant used for each language as follows:

- COBOL - 175.0
- PL/1 - 126.0
- Natural - 100.0
- Assembler - 400®.0

For for ADS Process or ADS Dialog resources, the **Backfired Function Points** value is not calculated.

Remember: The values are rounded before generating the report.

Project Inventory Report

This report presents a list of all the components in the current project.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Inventory Reports then double-click the desired report to display the report wizard.

In Report generation options area, the selection indicates that the report appendix is displayed by default. To show only the report summary, select the appropriate option.

The generated report has two sections: Report summary, which contains the total number of resources, grouped by type and Report details, which shows the detailed list of the resources that were found in the project.

Miscellaneous Reports

A-AUTO Network Dependency Report

This report shows for all selected networks the successor/predecessor networks, up to the selected level if specified.

To generate this report, in Explore Projects view, go to Mainframe Reports list in the pane to the right. Select Miscellaneous Reports, then double-click the desired report to display the report wizard.

In Report generation options area, choose the direction of the analysis (successors or predecessors). By default, the depth of the analysis is limited to three levels. The report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**.

The Report summary section presents the total number of input networks, the number of networks without successors/predecessors, and the length of the longest chain as well as a chart that shows Network dependency chain distribution.

The Report details section presents for each network the corresponding dependencies and the number of chains.

By default, an asterisk sign, "*", marks the chains for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all chains that reach the level limit set in the wizard are marked with an asterisk regardless of whether further analysis results are available or not. The meaning of the asterisk sign "*" in the context of each report is explained in a note at the end of the report.

Backward Call Chains Report

This report shows all call chains within the application, going backward with connectivity direction. For each program, the number of chains and the maximum length are determined.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Miscellaneous Reports, then double-click the desired report to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents a chart showing the Program call chain distribution.

The Report details section presents for each program, the number of chains per program.

Restricted nodes are be marked with a "!" sign. Restrictions can be defined from Preferences window, for details on how to set these restrictions see Call graph restrictions. By default, an asterisk sign, "*", marks the chains for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all chains that reach the level limit set in the wizard are marked with an asterisk regardless of whether further analysis results are available or not. The meaning of the asterisk sign "*" in the context of each report is explained in a note at the end of the report.

Forward Call Chains Report

This report shows all call chains within the application, going forward with connectivity direction. For each program, the number of chains, and the maximum length are determined.

To generate this report, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Select Miscellaneous Reports, then double-click the desired report to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window.

Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents a chart that shows the Program call chain distribution.

The Report details section presents for each program, the number of chains per program.

Restricted nodes are marked with a "!" sign. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

By default, an asterisk sign, "*", marks the chains for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limi**t tab from Preferences window). If **highlight results...** option is not checked, all chains that reach the level limit set in the wizard are marked with an asterisk regardless of whether further analysis results are available or not. The meaning of the asterisk sign "*" in the context of each report is explained in a note at the end of the report.

Job Flow Report

The Job Flow Report presents the job steps in the selected JCL or JCLs.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Miscellaneous Reports then double-click the desired report to display the report wizard.

The Available Job files area displays the job files available for report generation. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the job files between the Available and Selected lists then click **Finish** to generate the report.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**.

Mainframe Runtime Performance Report

This report traces the runtime performance for the Mainframe elements like: Jobs, Programs, CICS Transactions.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Miscellaneous Reports then double-click the desired report to display the report wizard.

In the report wizard dialog window, select the resource types for which you want to generate the report, then a date range, and click **Next**.

In the subsequent dialog window, the Available resources area lists the available resources according to the type selected in the previous dialog of the wizard. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the arrow buttons to move resources between the Available and the Selected lists. Once you selected the resources for which you want to generate the report, click **Finish**.

The report presents a list of resources with their corresponding runtime performance values for the selected time interval.

Rule Based Analysis Report

This report allows the user to select the coding rules to be used when generating the Rule Based Analysis report. The rules are stored in a special file, which can be modified/updated according to needs. For details on the specific syntax and an example of a rules file, see Appendix 2: Rules File Example for Rule Based Analysis.

To generate this report, in Explore Projects view, go to the Mainframe Reports list in the pane to the right. Select Miscellaneous Reports then double-click the desired report to display the report wizard.

In the report wizard dialog window, the Available rules area lists the coding rules available in the selected rules file. Use the **Name** field to filter the contents of the list. Select a coding rule from the list to display the rule description and the associated parameters (if applicable) in the Settings area. The parameters as well as the Weight for each coding rule can be customized. In Report generation options area, the selection indicates that the report appendix is displayed by default. Use the arrow buttons to move rules between the Available to the Selected lists then click **Next**.

The Available resources area lists the available resources. Use the **Name** field to filter the contents of the list. Once you selected the resources for which you want to generate the report, click **Finish**.

The Report summary section presents the list of rules that were used to generate the report and the list of resources that failed to comply with the selected rules. The rules names offer hyperlinks to the details section of the report.

The Report details section presents the name of the coding rule, the rule description, and category. Also, the report presents the list of resources that failed to comply with this rule. Furthermore, for each resource, the report presents the number of times the rule was broken and the line of code where the breach occurred. The number of the line of code where the breach occurred offers hyperlinks to the actual place in the resource. Clicking the hyperlink opens the resource in the editor at the line where the breach was identified.

SQL Backward Call Chains Report

This report shows the dependency SQL tables and for programs, all call chains within the application, going backward with connectivity direction. For each program, the SQL table(s) usage and the number of chains are determined.

To generate this report, in Explore Projects view go to the Mainframe Reports list in the pane to the right. Select Miscellaneous Reports then double-click the desired report to display the report wizard.

The Available SQL Tables area lists the SQL tables available in the selected project. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move resources between the Available to the Selected lists. Once you selected the resources for which you want to generate the report, click **Finish**.

The Report summary section presents the total number of programs, the number of programs never called, and the longest chain.

Restricted nodes are marked with a "!" sign. Restrictions can be defined from Preferences window, for details on how to set these restrictions, see Call graph restrictions.

By default, an asterisk sign, "*", marks the chains for which further analysis results are not displayed because of the limit set in the wizard. (**highlight results that are cut short by the threshold** option checked in **Analysis Expansion Limit** tab from Preferences window). If **highlight results...** option is not checked, all chains that reach the level limit set in the wizard are marked with an asterisk regardless of whether further analysis results are available or not. The meaning of the asterisk sign "*" in the context of each report is explained in a note at the end of the report.

Where Used Reports

Where Used Reports Overview

Where Used Reports show all the places within the application where specific resources are referenced. Resources are entities that are used by programs for I/O actions. For example, resources can be IDMS records or sets, VSAM files, BMS screens. The list of Where Used Reports depends on the environment in which the mainframe project was created and also on the resource types that exist within the current project.

For all resources, excluding variables and string literals, two types of reports can be generated: resource to program and program to resource. A resource to program report allows the selection of specific

resources and lists all programs that use those resources. A program to resource report allows the selection of specific programs and lists all resources that were used within these programs.

Adabas-Programs Report
This report shows for each Adabas resource, all the programs that use it.

To generate this report, go to Explore Projects view and select Mainframe Reports list. Select Where Used Reports, then double-click the desired report to display the report wizard.

The Available Adabas area lists the resources available for report generation in the selected project. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Adabas usage ratio and the Adabas distribution per programs.

The Report details section shows for all selected Adabas, all the programs that use the selected Adabas resources.

DAL-Programs Report
This report shows for each DAL resource, all the programs that use it.

To generate this report, goto **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available DAL area lists the resources available for report generation in the selected project. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the DAL usage ratio and the DALs distribution per programs.

The Report details section shows for all selected DALs, all the programs using the selected DAL resources.

Datacom-Programs Report
This report shows for each Datacom table, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Datacom Tables area lists the resources available for report generation in the selected project. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Datacom tables usage ratio and the Datacom tables distribution per programs.

The Report details section shows for all selected Datacom Tables, all the programs that use the selected tables.

Datasets-Jobs Report
This report shows for each Dataset, all the jobs where the selected dataset is defined and further used.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Datasets area lists the resources available for report generation in the selected project. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what extra data to include in the appendix: **show**

input details in appendix. Use the arrow buttons to move resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Datasets usage ratio and the Datasets distribution per jobs.

The Report details section shows for all selected Datasets, all the jobs where the selected dataset is defined. Also, for each job, a list of all the entities that are accessing it is presented. Furthermore, for each entity, the corresponding type is presented together with the number of occurrences.

Files-Programs Report
This report shows for each file, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Files area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Access type distribution and the Statement occurrences per file.

The Report details section shows for all selected files, all the programs that use those files.

IDMS Records-Programs Report
This report shows for each IDMS Record, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available IDMS Records area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows statement occurrences in programs is displayed for each IDMS record. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Access type distribution and the Statement occurrences per IDMS record.

The Report details section shows for all selected IDMS Records, all the programs that use those records.

IDMS Sets-Programs Report
This report shows for each IDMS Set, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available IDMS Sets area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows statement occurrences in programs is displayed for each IDMS set. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Access type distribution and the Statement occurrences per IDMS set.

The Report details section shows for all selected IDMS Sets, all the programs that use those sets.

IMS DB-Programs Report
This report shows for each IMS/DB resource, all the programs that use it, at IMS database segment level.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available IMS/DB area displays the IMS databases available for analysis. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Select the IMS/DB you want to analyze then click **Next**. A list of segments identified in the selected IMS database is presented. Select the segment or segments that you want to analyze.

In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows statement occurrences in programs is displayed for each IMS/DB. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Access type distribution and the Statement occurrences per IMS/DB.

The Report details section shows for all selected IMS/DB, all the programs that use those databases grouped by the accessed segment, whether directly or indirectly, that is, through the corresponding logical segment.

Includes-Programs Report
This program shows for each Include, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Includes area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents the total number of includes, the number of programs that access includes, the maximum number of distinct programs that access one include, the minimum number of distinct programs that access one include, and the average number of distinct programs that access one include followed by a chart that shows the Include distribution per programs.

The Report details section shows for all selected includes, all the programs that use those includes.

Jobs-Datasets Report
This report shows for each job, all the datasets that are used by the selected job.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Jobs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what extra data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that present the Datasets usage ratio and the Used datasets distribution.

The Report details section presents for each selected job the list of all used datasets. Furthermore, for each dataset the report presents the list of specific programs that access it. For each specific program, the corresponding type is presented. If it is confirmed that the corresponding dataset statement is used in that program, the number of occurrences is also displayed.

Jobs-Networks Report
This report shows for each selected job, all the networks that include the selected job.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Jobs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report details section presents for each selected job, the list of all networks that include that job.

Jobs-SQL Tables Report
This report shows for each job, all the SQL tables accessed based on the usage of the related programs.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Jobs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts presenting SQL tables usage ratio and Used SQL tables distribution.

The Report details section presents for each job the SQL tables that are accessed through the selected job as well as the name of the entity using the job to access the SQL table and the number of occurrences.

MQueues-Programs Report
This report shows for each MQueue resource, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Jobs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Programs usage ratio and the MQueues distribution per programs.

The Report details section shows for all selected MQueues, all the programs that use those MQueues.

Networks-Jobs Report
This report shows all the jobs included in a selected network.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available networks area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the jobs usage ratio and the network distribution per programs.

The Report details section shows for all selected networks, all the jobs included in those networks.

Programs-Adabas Report
This report shows for each program, all Adabas files used.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Adabas usage ratio and the Used Adabas Distribution.

The Report details section shows for all selected programs, all Adabas files used.

Programs-DAL Report
This report shows for each program, all DAL files that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the DAL usage ratio and the Used DAL Distribution.

The Report details section shows for all selected programs, all DAL files used.

Programs-Datacom Report
This report shows for each program, all Datacom tables used.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input**

details in appendix. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Datacom tables usage ratio and the Used Datacom tables Distribution.

The Report details section shows for all selected programs, all Datacom used.

Programs-Files Report
Use this report to generate results on all files used for each program.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The **Available Programs** area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the **Files usage ratio** and the **Used files distribution**.

The Report details section shows for all selected programs, the logical names of all files used.

Programs-IDMS Records Report
This report shows for each program, all IDMS records accessed.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows statement occurrences in programs is displayed for each IDMS record. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the IDMS records usage ratio and the Used IDMS records distribution.

The Report details section shows for all selected programs, all records accessed.

Programs-IDMS Sets Report
This report shows for each program, all IDMS Sets accessed.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows statement occurrences in programs is displayed for each IDMS set. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the IDMS sets usage ratio and the Used IDMS sets distribution.

The Report details section shows for all selected programs, all IDMS sets accessed.

Programs-IMS DB Report
This report shows for each program, the name of the IMS/DB accessed.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows statement occurrences in programs is displayed for each IMS/DB. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the IMS/DB usage ratio and the Used IMS/DB distribution.

The Report details section presents for each program: the name of the IMS/DB accessed, the segment, the type of statement and the statement occurrences.

Programs-Includes Report
This report shows for each program, all Includes accessed.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents the following data: the total number of programs, the number of accessed includes, the maximum number of includes accessed from a program, the minimum number of includes accessed from a program, and the average number of includes accessed from a program. Also, the summary presents two charts that show Programs accessing includes usage ratio and the Include distribution per file.

The Report details section shows for all selected programs, all includes accessed.

Programs-Literals Report
This report shows for each program, all string literals.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows literals

occurrences is displayed for each program. Use the arrow buttons to move the resources between the Available and Selected lists then click **Next**.

In the second dialog window of the report wizard, a list of search patterns is displayed. Use **Add** to add new search patterns to the list. Each keyword can contain wildcard characters, use **Remove**, or **Remove all** to clear the list of search criteria. Click **Finish** button to start generating the report.

The Report summary section presents two charts that show the Literals usage ratio and the Used literals distribution.

The Report details section shows for all selected programs, all string literals.

Programs-MQueues Report
This report shows for each program, all the MQueues accessed.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Next** to generate the report.

The Report summary section presents two charts that show the MQueues usage ratio and the Used MQueues distribution.

The Report details section presents for each program: the MQueue accessed, the type of statement and the statement occurrences.

Programs-SQL Tables Report
This report shows for each program the name of the SQL table that is referenced, the type of statement and the statement occurrences.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Next** to generate the report.

The Report summary section presents two charts that show the SQL tables usage ratio and the Used SQL tables distribution.

The Report details section presents for each program: the name of the SQL table that are referenced, the type of statement and the statement occurrences.

Programs-Variables Report
This report shows for each program, all variables.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available Programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button

to the left of the dialog window. Use the arrow buttons to move the resources between the Available and Selected lists. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. If you deselect the option **hide detailed images**, a chart that shows variables occurrences is displayed for each program.

Click **Next** to display the second dialog window of the report wizard: A list of search patterns is displayed. Use **Add** to add search patterns to the list. Each keyword can contain wildcard characters, use **Remove** or **Remove all** to clear the list of search criteria. Click **Finish** to start generating the report.

The Report summary section presents two charts that show the Variables usage ratio and the Used variables distribution.

The Report details section shows for all selected programs, all variables.

SQL Tables-Jobs Report
This report shows for each SQL table, all the jobs that access it based on the usage of the related programs for those jobs.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available SQL Tables area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show SQL tables usage ratio and SQL table distribution per jobs.

The Report details section presents for each SQL table, all the jobs based on the usage of the related programs.

SQL Tables-Programs Report
This report shows for each SQL Table, all the programs that use it.

To generate this report, go to **Explore Projects** view and select **Mainframe Reports** list. Select **Where Used Reports**, then double-click the desired report to display the report wizard.

The Available SQL Tables area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. In Report generation options area, the selection indicates that the report appendix is displayed by default. Choose what additional data to include in the appendix: **show input details in appendix**. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the SQL tables usage ratio and the SQL tables distribution per programs.

The Report details section shows for all selected SQL Tables, all the programs that use those SQL tables.

Shared Resources Report

To generate the **Shared Resources Report**, in **Application Discovery Browser** perspective, from the **Projects** list in the central part of the **Explore Projects** view, select the projects for which you want to generate a **Shared Resources Report**. For details on how to select a perspective and a description of **Application Discovery Browser** perspective, see IBM AD Analyze Client Reference.

To generate this report, goto **Shared Resources** section, and double-click **Shared Resources Report** to display the **Shared Resources Report** wizard dialog window.

The Available types area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list.

Use the arrow buttons to move elements between the Available and Selected lists then click **Finish** to generate the report.

Compare Projects Report

The compare operation can be performed on two or more projects. For mainframe projects, the comparison can be performed only between two projects.

To generate the Compare Projects Report, go to the **Application Discovery Browser** perspective, and select the projects for which you want to generate a Compare Projects report. For details on how to select a perspective and a description of the **Application Discovery Browser** perspective, see section IBM AD Analyze Client Reference.

In the Compare Analysis section, to the right of the Explore Projects view, double-click Compare Projects Report to generate the report.

The report summary presents a list with the total numbers of elements that were Added, Changed, and Removed, grouped by category. Then, under the Added, Changed, and Removed sections the report presents the detailed list of resources identified as such.

Example

To understand how a Compare Projects Report interprets the added, changed, or removed resources, consider the following assumption:

Project1
Contains COBOL for z/OS resources A, B, and C.

Project2
Contains COBOL for z/OS resources B, C, and X.

COBOL for z/OS resource A
Contained only in Project1.

COBOL for z/OS resource B
Contained in both Project1 and Project2. This resource has identical content at the file level, which means the source is unchanged.

COBOL for z/OS resource C
Contained in both Project1 and Project2. This resource has the content changed. The COBOL program C from Project2 has two statements additionally added compared with the COBOL program C from Project1.

COBOL for z/OS resource X
Contained only in Project2.

In the Compare Projects Report, projects are shown alphabetically no matter in what sequence the projects are selected. When selecting both Project1 and Project2, Project1 is shown on the left, while Project2 is shown on the right. The comparison term is considered Project1. Therefore, the report contains the following results:

- The Added section prints the A COBOL program, which is included only in Project1.
- The Changed section prints the C COBOL program, which is contained in both Project1 and Project2, but this program has the content changed at the file level.
- The Removed section prints the X COBOL program, which is not added in Project1 but contained in Project2.
- The common resources that have the same content in both projects, for example, the B COBOL program, are not printed at all in this report.

Legacy Java Reports

Legacy Java Complexity Reports

Legacy Java Complexity Report
This report presents complexity measurements developed to measure the complexity of a class directly from source code.

To generate this report, goto **Explore Projects** > **Java Reports list** > **Complexity Reports** and double-click the desired report to display the report wizard.

The Available programs area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Place the cursor over an icon to display a tooltip about the type of resource it displays and click it to display only the selected type of resource. To display the full name of the resource type, click the white arrow button to the left of the dialog window. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Complexity percentage distribution and the Files complexity distribution as well as a Complexity score table for all the selected resources.

The Report details section shows for each selected resource: the resource name, the McCabe value calculated for it, and the corresponding complexity value.

McCabe Complexity Report
The McCabe Complexity Report for Java projects presents Cyclomatic complexity (a broad measure of soundness and confidence for a class) for each project entity. Cyclomatic complexity measures the number of linearly-independent paths through a class. This report presents the complexity percentage distribution graph and file complexity distribution graph for the project.

Note: The settings for the McCabe complexity report are available in the Preferences dialog window. For details, see McCabe Complexity Report Java.

To generate this report, goto **Explore Projects** > **Java Reports list** > **Complexity Reports** and double-click the desired report to display the report wizard.

The Available resources area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the Complexity percentage distribution and the Files complexity distribution as well as a Complexity score table for all the selected resources.

The Report details section shows for each selected resource: the resource name, the **McCabe** value calculated for it and the corresponding complexity value.

Inventory Reports for Legacy Java Projects

Legacy Java Dead Code Report
This report lists all Java methods that are not referenced by other methods within the project.

These methods must be tested to determine whether they are main methods (and therefore are not referenced) or whether they are not used and can be considered dead code.

To generate this report, goto **Explore Projects** > **Java Reports list** > **Inventory Reports** and double-click the desired report to display the report wizard.

The Report summary section presents the list of unused methods that were found.

Legacy Java Inventory Report
This report presents a list of all the components in the current project.

To generate this report, goto **Explore Projects** > **Java Reports list** > **Inventory Reports** and double-click the desired report to display the report wizard.

The generated report presents a summary with the total number of resources grouped by type, followed by the detailed list of the resources found in the project.

Third-Party Usage Report
The Third-Party Usage Report presents a list of the classes that use third-party entities (classes and packages).

To generate this report, goto **Explore Projects** > **Java Reports list** > **Inventory Reports** and double-click the desired report to display the report wizard.

The Available resources area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

Legacy Java Where Used SQL Table
This report shows for each SQL Table, all the Java resources that use it.

To generate this report, goto **Explore Projects** > **Java Reports list** > **Java Where Used SQL Table** to display the report wizard.

The Available SQL Tables area displays the resources available for report generation and allows the user to select the resources to be included in the report. Use the **Name** field to filter the contents of the list. Use the arrow buttons to move the resources between the Available and Selected lists then click **Finish** to generate the report.

The Report summary section presents two charts that show the SQL tables usage ratio and the SQL tables distribution per Java resource.

The Report details section shows for all selected SQL Tables, all the Java resources that use those SQL tables.

Usage

Usage in Jobs

Dataset Usage in Jobs
Presents the Dataset Usage in Jobs analysis available in **IBM AD Analyze Client**.

Dataset Usage in Jobs tab shows defined datasets within project sources.

To display **Dataset Usage in Jobs** tab, go to **Explore Projects tab** and select **Dataset Usage in Jobs** from the **Usage in Jobs** list of analyses to the right of the screen.

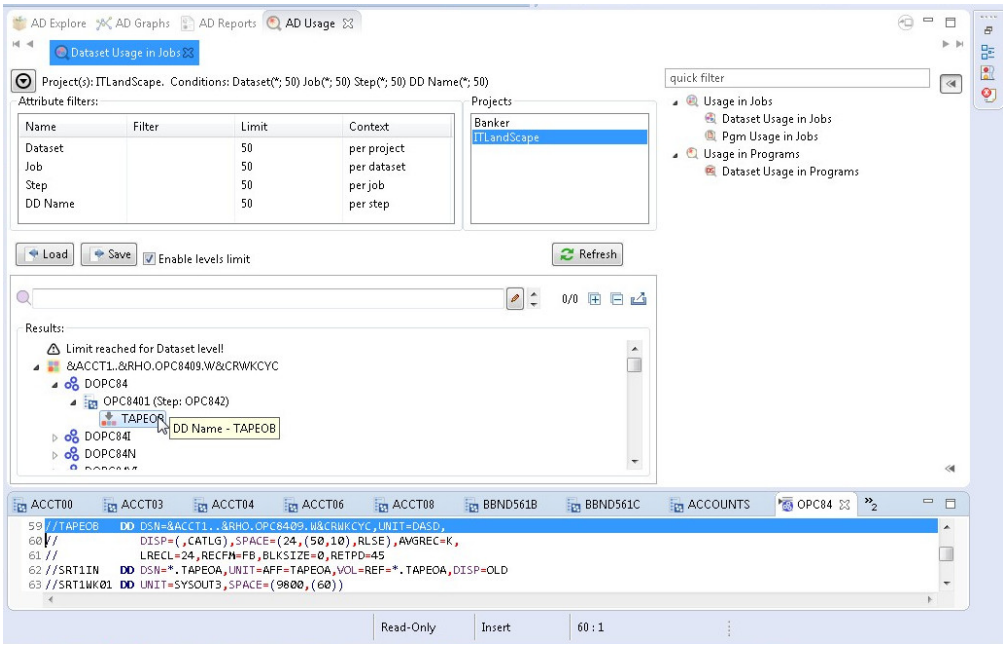


Figure 59. Dataset Usage in Jobs

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Dataset Usage in Jobs Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list.

To the right of Attribute filters area, the list of projects is displayed. Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. After the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Dataset Usage in Jobs parameters in a separate tab, use the arrow button in the upper right corner of the **Dataset Usage in Jobs** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button, **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area the following buttons are available:

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 18. Dataset Usage in Jobs tab nodes

Node name	Description
Dataset	Shows the list of occurrences in the JCL where the Data Set is allocated for the specific logical file. ”Unknown” label is displayed for Dataset with unknown name.
Job	Shows the job where the selected dataset is used.
Program/step	Shows the name of the program that is executed in step.
DD Name	Displays the DD name that is associated with the data set.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Job Usage Inventory

Job Usage Inventory tab shows the structure of the JCL listing the defined datasets, executed procedures, and programs.

To display **Job Usage Inventory** tab, in **Explore Projects** tab double-click Job Usage Inventory from the Usage in Jobs list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Job Usage Inventory Settings. The Context column displays the context of the filter operation where relevant.

To the right of Attribute filters area, the list of projects is displayed. Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. After the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Job

Usage Inventory parameters in a separate tab, use the arrow button in the upper right corner of the **Job Usage Inventory** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area the following buttons are available:

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 19. Job Usage Inventory Tab nodes

Node name	Description
Jobs	Shows the list of jobs that were found in the resources of the selected project.
Datasets	Lists the datasets defined in JCL.
Procedures	Lists the procedures that were identified in the job.
Programs	Lists the programs that are executed in JCL.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Pgm Usage in Jobs

The **Pgm Usage in Jobs** tab shows the programs to be executed in jobs.

To display **Pgm Usage in Jobs** tab, in **Explore Projects** tab double-click Pgm Usage in Jobs from the Usage in Jobs list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Pgm Usage in Jobs Settings. The Context column displays the context of the filter operation where relevant. The context

of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list.

To the right of Attribute filters area, the list of projects is displayed. Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Pgm Usage in Jobs parameters in a separate tab, use the arrow button in the upper right corner of **Pgm Usage in Jobs** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button, **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area the following buttons are available:

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. Once you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 20. Pgm Usage in Jobs Tab nodes

Node name	Description
Program	Lists the programs or Load Modules or Utilities executed in jobs.
Job	Lists the jobs where programs or Load Modules or Utilities are used.
Step	Shows the step where the DD library is mentioned.
DD Name	Shows the name of the DD library.
Dataset	Shows the dataset.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Procedure Usage in Jobs

The **Procedure Usage in Jobs** tab shows the procedures invoked by jobs.

To display the **Procedure Usage in Jobs** tab, in **Explore Projects** tab double-click Procedure Usage in Jobs from the Usage in Jobs list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Procedure Usage in Jobs Settings.

The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list.

To the right of Attribute filters area, the list of projects is displayed. Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Procedure Usage in Jobs parameters in a separate tab, use the arrow button in the top right corner of **Procedure Usage in Jobs** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area the following buttons are available:

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip presenting the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 21. Procedure Usage in Jobs nodes

Node name	Description
Procedure	Lists the procedures that were identified in the resources of the selected project.
Job	Shows the job where the procedure is invoked.
Step	Shows the step where the DD library is mentioned.
DD Name	Shows the name of the DD library.
Dataset	Shows the dataset name.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node that is selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Usage in Programs

Adabas Usage

The **Adabas Usage** tab presents how the Adabas is being used throughout the programs, such as Natural and Cobol.

To display the **Adabas Usage** tab, in **Explore Projects** tab double-click **Adabas Usage** from the Usage in Programs list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding **Usage Settings** section from **Preferences**. For more information, see Adabas Usage Settings. The Context column displays the context of the filter operation where relevant. To modify the attributes' order in the list use the green arrow buttons to the left of the attributes list. When changing the attributes' order, the context of the filtering operation changes accordingly.

To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the **Adabas Usage** parameters in a separate tab, use the arrow button in the upper right corner of **Adabas Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button, **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click a paragraph, variable, or include to open the corresponding file at the occurrence line.

The available nodes depend on the type of Adabas resource selected. If the Adabas resource is accessed by Natural programs, the following nodes are available:

Table 22. Adabas Usage Tab nodes - Natural

Node name	Description
Adabas DB	Shows the Adabas DB ID.
Adabas File	Shows the Adabas file number.
Adabas View	Presents the view that is used to see the statement. This option can be used to check in which view or views each statement is used. The View name is presented as a concatenation of the following elements: the name of Adabas View, the name of Natural DB View, and the resource name (the Data Area name).
Program	Opens the program file at the Program ID row.
Statement type	Displays the statement types.
Statement Source Occurrences	Opens the program at the line that contains the selected statement.

If the Adabas resource is accessed by Cobol programs, the following nodes are available.

Table 23. Adabas Usage Tab nodes - Cobol

Node name	Description
Adabas DB	Shows the Adabas DB ID.
Adabas File	Shows the Adabas file number.
Adabas View	Presents the view that is used to see the statement. This option can be used to check in which view or views each statement is used. The View name is presented as N/A (not available) for Cobol programs with Adabas direct call and as Adabas view name for Cobol programs with Adabas Native SQL.
Program	Opens the program file at the Program ID row.
Statement type	Displays the statement types.
Statement Source Occurrences	Opens the program at the line that contains the selected statement.

Adabas Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **Adabas Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **Adabas Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Assembler Program Structure

The **Assembler Program Structure** tab presents the structure of the Assembler programs in the current project.

To display the **Assembler Program Structure** tab, go to **Explore Projects** tab and select **Assembler Program Structure** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Assembler Program Structure Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Assembler Program Structure parameters in a separate tab, use the arrow button in the upper right corner of **Assembler Program Structure** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Expand top 4 levels, click this button to expand the top four levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 24. Assembler Program Structure tab nodes	
Node name	**Description**
Assembler Program	Lists the tree for an Assembler source program that contains nodes for CSECTs, ENTRYs, includes, and macros.
CSECT	Lists the CSET sections that were defined in Assembler.
ENTRY	Lists the ENTRY sections that were defined in Assembler.
Macro	Lists Assembler Macro that is used in Assembler.
Includes	Lists includes used in Assembler.

Assembler Macro Expansion

Important: The **Assembler Macro Expansion** feature is an extension of the base Assembler compiler and it is not enabled by default. To enable the **Assembler Macro Expansion** see the "Enabling the Assembler Macro Expansion Feature" on page 238 section.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Assembler Variable Usage

The **Assembler Variable Usage** tab presents the usage of variables in the Assembler programs in the current project.

To display the **Assembler Variable Usage** tab, go to **Explore Projects** tab and select **Assembler Variable Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Assembler Variable Usage Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Assembler Variable Usage parameters in a separate tab, use the arrow button in the upper right corner of **Assembler Variable Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Expand top 4 levels, click this button to expand the top four levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 25. Assembler Variable Usage Tab nodes

Node name	Description
Assembler program	Lists the tree for an Assembler source program that contains nodes for CSECTs, ENTRYs, and variables.
CSECT	Lists the CSET sections that are defined in Assembler.
ENTRY	Lists the ENTRY sections that are defined in Assembler.
Variables	Lists all the variables in the Assembler file.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Cobol Program Structure

The **Cobol Program Structure** tab shows the structure of Cobol programs by paragraph and copybook usage.

To display the **Cobol Program Structure** tab, go to **Explore Projects** tab and select **Cobol Program Structure** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Cobol Program Structure Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Cobol Program Structure parameters in a separate tab, use the arrow button in the upper right corner of **Cobol Program Structure** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Expand top 2 levels, click this button to expand the top two levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click a paragraph or include to open the corresponding file at the occurrence line. The following nodes are available:

Table 26. Cobol Program Structure Tab nodes

Node name	Description
COBOL file	Lists the tree for a COBOL source file that contains nodes for paragraphs and include files.
Paragraphs	Lists the paragraphs in the COBOL file.
Includes	Lists the COPY or INCLUDE statement or statements within the program source code.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Cobol Variable Usage

The **Cobol Variable Usage** tab shows variable usage in COBOL programs and also in User-Defined Functions (that are currently treated as COBOL programs), identifies the variables that are declared in COBOL Copybooks/INCLUDES, and displays the Include names within ( ) brackets. The variables that are defined in the program source will not have the ( ) brackets.

To display the **Cobol Variable Usage** tab, go to **Explore Projects** tab and select **Cobol Variable Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"

- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Cobol Variable Usage Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements which are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Cobol Variable Usage parameters in a separate tab, use the arrow button in the top right corner of **Cobol Variable Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Expand top 2 levels, click this button to expand the top two levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip presenting the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click a variable to open the corresponding file at the occurrence line. The following nodes are available:

Table 27. Cobol Variable Usage Tab nodes

Node name	Description
COBOL file	Lists the tree for a COBOL source file that contains nodes for variables.
Variables	Lists all the variables in the COBOL file.

Cobol Variable Usage pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

DAL Usage

The **DAL Usage** tab shows the access to the DAL files within the Cobol programs in the current project. For each DAL file, the programs and the statement types that reference the file are shown and for each statement type, the occurrences of the statement (the file and location within the file) are listed.

To display the **DAL Usage** tab, go to **Explore Projects** tab and select **DAL Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see DAL Usage in Program Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the DAL Usage parameters in a separate tab, use the arrow button in the upper right corner of **DAL Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand tree branches.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

The following nodes are available:

Table 28. DAL Usage Tab nodes

Node name	Description
DAL file	Presents a list of the DAL files in the project. The information in the sub-tree under the DAL file represents where used information for the DAL file.
Program	Opens the selected program at the first line.
Statement type	This node represents the access type that is used to access the DAL file. If the access type is dynamic and cannot be resolved in the analysis stage, the access type “DAL: Dynamic_Function” is shown. Child nodes show the name of the source file and the location (line number) of the DAL function.

DAL Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **DAL Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **DAL Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Datacom Table Usage

The **Datacom Table Usage** tab lists the Datacom tables that are used by the application. Nodes in the Datacom tab represent the different statement types that refer to the tables (ADDIT, LOCKX, UPDAT). For each statement type, the references in the source code (file name and location of the reference) are listed.

To display the **Datacom Table Usage** tab, go to **Explore Projects** and select **Datacom Table Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes in the set order together with the filters set for each attribute

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of elements to be returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Datacom Table Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Dataset Usage in Programs parameters in a separate tab, use the arrow button in the upper right corner of **Datacom Table Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 29. Datacom Table Usage Tab nodes

Node name	Description
Datacom table	Shows the list of Datacom tables that are accessed.
Program	Presents the list of programs that access the Datacom table.
Statement type	Displays the statement type.
Statement source info	Opens the program at the line that contains the selected statement.

Datacom Table Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **Datacom Table Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **Datacom Table Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Dataset Usage in Programs

The Dataset Usage in Programs analysis gives a view of the application from the Data Sets point of view. For each physical Data Set used, you can see the places where it is referred to within the JCL jobs, the places where it is used in the COBOL programs, and also the way it is used by different programs.

To display the **Dataset Usage in Programs** tab, go to **Explore Projects** and select **Dataset Usage in Programs** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes in the set order together with the filters set for each attribute

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of elements to be returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Dataset Usage in Programs Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Dataset Usage in Programs parameters in a separate tab, use the arrow button in the upper right corner of **Dataset Usage in Programs** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Click **Expand all** to expand the tree.

Click **Collapse all** to fold all the tree branches.

Use **Export Results to CSV File** to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 30. Dataset Usage in Programs Tab nodes

Node name	Description
Dataset	Shows the list of occurrences in the JCL where the Dataset is allocated for the specific logical file. ”Unknown” label is displayed for dataset with unknown name.
DD name	Displays the DD name that is associated with the dataset.

Table 30. Dataset Usage in Programs Tab nodes (continued)	
Node name	**Description**
Logical name	Displays the logical file name that is assigned to the physical file name.
Program category	Displays the program category.
Program	Presents the list of programs that use the Dataset. Gathered information includes calls between programs. For example, if a main program that is called by the job invokes a subprogram that uses a dataset, that information is also included in the list.
Statement type	Displays the statement type.
Statement source info	Opens the program at the line that contains the selected statement.

Dataset Usage in Programs statement filtering

Starting with **IBM AD V5.1.0.9**, the **Dataset Usage in Programs** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **Dataset Usage in Programs** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

IDMS Usage

IDMS Usage tab shows the usage of IDMS sets and records within the IDMS Schema that is used by Cobol programs.

To display the **IDMS Usage** tab, go to **Explore Projects** tab and select **IDMS Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes in the set order with the filters set for each attribute

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of elements to be returned by the search, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see IDMS Usage Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the

attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the IDMS Usage parameters in a separate tab, use the arrow button in the upper right corner of **IDMS Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Reorder current results, use this button to order the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand top 5 levels, click this button to expand the top five levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available:

Table 31. IDMS Usage Tab nodes

Node name	Description
Schema	Displays the name of Schema as parent for Subschema. "No Schema" label if schema does not exist.
Subschema	Displays the name of Subschema that is used by Record/Set in Cobol Program; or "No Subschema" label if subschema does not exist.
Record	Lists the used records.
Set	Lists the used sets.
Program	Displays the name of the Cobol Program where the Record or the Set is used.
Statement type	Displays the usage statement type.
Statement source info	Presents the source occurrences of the statement in the selected Program.

IDMS Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **IDMS Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **IDMS Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

IMS/DB Logical Usage

The **IMS/DB Logical Usage** tab shows the way the segments of the Logical IMS database are used across the programs in the project. Also, for each logical segment, the source physical segment and physical database are presented. Data that is contained in a segment can be grouped in one or more fields. The IMS/DB fields that are used to qualify a call statement are presented as part of the results.

IMS/DB is a hierarchical database (unlike a relational or a network one) and it is made out of segments and their relationships. A segment is similar to a record and it is the basic unit of data for IMS/DB. A segment can have many siblings but it can have only one parent segment. This limits the complexity of the physical database. To overcome this limitation, DL/I allows the database administrator to implement logical relationships in which a segment can have both physical and logical parents. Additional relationships can thus be created within one physical database. Once the logical relationships are implemented, the new structure is known as the Logical Database.

To display the **IMS/DB Logical Usage** tab, go to **Explore Projects** tab and select **IMS/DB Logical Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected project or projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding **Usage Settings** section from the **Preferences** dialog. For more information, see “IMS/DB Logical Usage Settings” on page 442 . The Context column displays the context of the filter operation where relevant. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the IMS/DB Logical Usage parameters in a separate tab, use the arrow button in the upper right corner of **IMS/DB Logical Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search

results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an entry to open the corresponding file at the occurrence line. The following nodes are available:

Table 32. IMS/DB Logical Usage Tab nodes

Node name	**Description**
Logical database	Presents the logical IMS/DB name.
Logical segment	Presents the logical segment name.
Physical database name	Presents the physical IMS/DB name, the source database of the logical database.
Physical segment name	Presents the physical segment name, the source segment of the logical segment.
Program	Presents the program that contains the referred statement.
Statement type	Presents the statement type and source occurrences of the statement in selected program.
Search field	Presents the IMS/DB field that is specified as search criteria in the referred statement. The sequence field is identified by a distinct icon, which contains the *'k'* sign.

Physical Database and Physical Segment Name levels are related to Logical Segment level (representing the source database and source segment for logical segment).

Accordingly:

1. Physical Database and Physical Segment will always be under Logical Segment level (even if results level or filters order is changed).

2. If Logical Segment is disabled from filters, those 2 levels will not be displayed in result tree.

IMS/DB Logical Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **IMS/DB Logical Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **IMS/DB Logical Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes.

Note: If you select **View analysis source** for a Logical database node, the occurrence from the *.DBD file is presented in the Editor view.

According to the selected node, several **Collapse to ... Level** options might also be available.

IMS/DB Structure

The **IMS/DB Structure** tab presents the relationships among the segments and fields of an IMS database, grouped by access type (Fast path, Logical, Full function, and File access).

To display the **IMS/DB Structure** tab, go to **Explore Projects** tab and select **IMS/DB Structure** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected project or projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see “IMS/DB Structure Settings” on page 442. The Context column displays the context of the filter operation where relevant. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the IMS/DB Usage parameters in a separate tab, use the arrow button in the upper right corner of **IMS/DB Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs,

reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information about the selected element attributes collected from element definition.

The results of the IMS/DB Structure analysis are grouped by IMS/DB access type: Fast path, Logical, Full function, and File access.

In the tree, an entry to open the corresponding file at the occurrence line. The following nodes are available:

Table 33. IMS/DB Usage Tab nodes

Node name	Description
Database type	Presents the database type, this can be either physical or logical database.
Database	Presents the database name.
Segment	Presents the segment name, this can be either a physical or a logical segment.
Field	Presents the IMS/DB fields. The sequence (key) field is identified by a distinct icon, which contains the *'k'* sign.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes.

Note: View analysis source for all the nodes of this analysis will show occurrences in the *.DBD file.

According to the selected node, several **Collapse to ... Level** options might also be available.

IMS/DB Usage

The **IMS/DB Usage** tab shows the way the segments and fields of an IMS database are used across the programs in the project.

To display the **IMS/DB Usage** tab, go to **Explore Projects** tab and select **IMS/DB Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected project or projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding **Usage Settings** section from the **Preferences** dialog. For more information, see "IMS/DB Usage Settings" on page 442. The Context column displays the context of the filter operation where relevant. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply**

to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the IMS/DB Usage parameters in a separate tab, use the arrow button in the upper right corner of **IMS/DB Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element. The information regarding the logical database and logical segment name is displayed in the **Properties** view.

In the tree, an entry to open the corresponding file at the occurrence line. The following nodes are available:

Table 34. IMS/DB Usage Tab nodes

Node name	**Description**
Database type	Presents the database type, this can be either physical or logical database.
Database name	Presents the name of the database.
Segment name	Presents the IMS/DB segments.
Program	Presents the programs which contain the referred statement.
Statement type	Presents the statement type and source occurrences of the statement in selected program.
Statement source info	Presents the name of the resource where the statement was found.
Search field	Presents the IMS/DB field that is specified as search criteria in the referred statement. The sequence field is identified by a distinct icon, which contains the *'k'* sign.

IMS/DB Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **IMS/DB Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **IMS/DB Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Include Usage

The **Include** tab displays information on the usage of Includes in the project. The nodes represent the Includes in the project. Expanding an include node displays a list of nodes that represent Include statements in COBOL files that refer to the copybook. The name of the file that contains the Include statement and the statement's location (line and column numbers) are shown.

To display the **Include Usage** tab, go to **Explore Projects** tab and select **Include Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes in the set order with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of elements to be returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Include Usage Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Include Usage parameters in a separate tab, use the arrow button in the upper right corner of **Include Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Reorder current results, use this button to order the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand top 5 levels, click this button to expand the top five levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 35. Include Usage Tab nodes

Node name	Description
Include	Presents a list of the Include files in this project.
Program category	Lists the types of programs.
Program	Presents the include files that are used in programs: Cobol, Natural, PL/I, Assembler, ADS.
Statement type	Presents the statement type and source occurrences of the statement in selected program.

Include Usage variable filtering

Starting with **IBM AD V5.1.0.8**, the Include Usage analysis has been expanded with an extra layer of filtering and has been designed to locate more effectively specific variables from includes inside programs.

The new filtering option is available in the **Attribute Filters** box. By default, it is automatically unchecked and it needs to be selected to activate it.

Without the Variable filter option, you will be able to see only **Include** > **Program** > **Statement Type** > **Statement Source Info**. If you want to find a variable, you must search manually in the **View Analysis Source** box.

If the Variable filter is selected, an extra level of information is displayed and contains all variables from the include. You will be able to see **Include** > **Program** > **Statement Type** > **Statement Source Info** > **Variable**.

If you left-click on a specific variable, additional information is displayed in the **Variable Usage Properties** box and includes the **Start Row**, which makes it easier to find the target variable. You can also double-click the variable, or right-click and select **View Analysis Source**, and see the exact row where the variable is located.

In the **Filter** column, you can manually type any variable name and click on **Apply/Refresh** to display only the typed variables.

Note:

- You can use wildcards to enable a broader searching experience.
- After you click on a variable, all available analyses are displayed on the right side.

Include Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **Include Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **Include Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

MQueue Usage

The **MQ Usage** tab presents how the MQ Series is being used throughout the programs. When you send or receive a message using MQ, you need to specify the queue manager you are working with and the queue you are using. Therefore this tab has the following levels: queue manager, queue, program, statement type, source occurrences.

To display the **MQueue Usage** tab, go to **Explore Projects** tab and select **MQueue Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected project or projects, the selected attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see MQueue Usage Settings. The Context column displays the context of the filter operation where relevant. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the MQueue Usage parameters in a separate tab, use the arrow button in the upper right corner of the **MQueue Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area the following buttons are available.

Reorder current results, use this button to reorder the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click a paragraph, variable, or include to open the corresponding file at the occurrence line. The following nodes are available.

Table 36. MQueue Usage Tab nodes

Node name	Description
Queue	Presents a list of queues that are used in this project.
Queue Manager	Presents the name of Queue Manager.
Program	Presents the programs that contain the referred statement.
Statement type	Presents the statement type and source occurrences of the statement in selected program.

MQueue Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **MQueue Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **MQueue Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node several **Collapse to ... Level** options might also be available.

Natural Program Structure

The **Natural Program Structure** tab shows the subroutines and includes found in the Natural programs in the current project.

To display the **Natural Program Structure** tab, go to **Explore Projects** tab and select **Natural Program Structure** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute.

Important:

In AD Analyze, an external subroutine from a Natural file is treated as a program when it exists independently within a Natural program file and when it can be called from any other Natural program, except the one where it has been defined.

Note: Make sure that the external subroutine's name is unique within the application library.

An external subroutine has the following structure:

- Data Definition (optionally)

- Control logic (where the control logic is coded as a subroutine, e.g. starts with DEFINE SUBROUTINE and ends with END-SUBROUTINE)
- End Statement

Note: The control logic for external subroutines may contain multiple subroutine definitions nested within and should not have other logic outside subroutine's definition section.

Only in this case, an external subroutine will be treated as a program.

Therefore, the Natural Program Structure tab shows the subroutine name prefixed with the module name. Example:

```
module:ExternalSubroutine
```

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as follows:

  ```
   "AAA*" or "*AAA"
  ```

  or

  ```
  "module*" or "*ExternalSubroutine"
  ```

- "," is used to search multiple elements as in "AAA, BBB, CCC, module:ExternalSubroutine1, module:ExternalSubroutine2"
- "module:ExternalSubroutine" is used to directly search the name displayed in the list

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Natural Program Structure Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Natural Program Structure parameters in a separate tab, use the arrow button in the upper right corner of the **Natural Program Structure** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Expand top 2 levels, click this button to expand the top two levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 37. Natural Program Structure Tab nodes	
Node name	**Description**
Program	Shows the tree for a Natural source file. The tree contains nodes for subroutines and include files.
Subroutines	Opens the source file at the relevant subroutine.
Includes	Opens the source file at the include statement.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

Natural Variable Usage

The **Natural Variable Usage** tab shows the variable usage for the Natural programs in the current project.

Note: Starting with IBM AD V6.1.0 the external subroutines are treated as programs. For more information, see the external subroutines structure.

To display the **Natural Variable Usage** tab, go to **Explore Projects** tab and select **Natural Variable Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects and all attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see Natural Variable Usage Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the Natural Variable Usage parameters in a separate tab, use the arrow button in the upper right corner of the **Natural Variable Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Expand top 1 level, click this button to expand the first level of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 38. Natural Variable Usage Tab nodes

Node name	Description
Program	Shows the tree for a Natural source file. The tree contains nodes for variables.
Variables	Opens the source file at the relevant Variable declaration. Expands the Variable node to show how a variable is used, divided into Read (where the variable value is read) and Write (where the variable value is written).

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

PL/I Program Structure

The **PL1 Program Structure** tab shows the procedures and include files for the PL/I programs in the current project.

To display the **PL1 Program Structure** tab, go to **Explore Projects** tab and select **PL1 Program Structure** from the **Usage in Programs** list of analyses to the right of the screen.

Important: In AD Analyze, external procedures from a PL/I file are treated as programs for the following cases:

- In case *package exports* statement exists in the PL/I file only, the procedures listed in this statement are seen as external procedures, namely programs.
- In case *package exports* statement doesn't exists in the PL/I file, all the procedures defined on the first level in the file are seen as external procedures, namely programs.

Therefore the PL1 Program Structure tab shows the procedure name prefixed with the member name (example: **member:procedure**).

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes together with the filters set for each attribute. Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. When you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used; for example, you can use the special character "*" and add to it either prefixes or suffixes as in **Member*** or ***Procedure**. In order to filter right after the name displayed in the list, you can use the **member:procedure** notation.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding **Usage Settings** section from the **Preferences** dialog. For details, see PL/I Program Structure Settings.

The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the PL/I Program Structure parameters in a separate tab, use the arrow button in the top right corner of the **PL1 Program Structure** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Expand top 2 levels, click this button to expand the top two levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 39. PL/I Program Structure Tab nodes

Node name	Description
Program	Shows the tree for a PL/I source file. The tree contains nodes for procedures and include files.
Procedure	Opens the source file at the relevant procedure.
Includes	Opens the source file at the include statement. **Note:** For a copybook that is included outside of any PL/I program in the source file, the copybook is displayed in the result tree of every PL/I program from that file.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

PL/I Variable Usage

The **PL1 Variable Usage** tab shows the variable usage for the PL/I programs in the current project.

To display the **PL1 Variable Usage** tab, go to **Explore Projects** tab and select **PL1 Variable Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. The breadcrumb displays the selected projects and all attributes together with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"

- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of attributes of a certain type that are returned by the filtering operation, use the fields in the Limit column. The default limit is 10; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see PL/I Variable Usage Settings. The Context column displays the context of the filter operation where relevant. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the PL/I Variable Usage parameters in a separate tab, use the arrow button in the upper right corner of the **PL1 Variable Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Expand top 2 levels, click this button to expand the top two levels of the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 40. PL/I Variable Usage Tab nodes

Node name	Description
Program	Shows the tree for a PL/I source file. The tree contains nodes for variables.
Variables	Opens the source file at the relevant Variable declaration. Expands the Variable node to show how a variable is used, divided into Read (where the variable value is read) and Write (where the variable value is written). **Note:** For a global variable, only if it is used in a PL/I program, the global variable is displayed in the result tree of the corresponding program.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

SQL Table Field Usage

The **SQL Table Field Usage** tab shows the usage of table fields of the SQL tables, within the programs in the current project.

To display the **SQL Table Field Usage** tab, go to **Explore Projects** tab and select **SQL Table Field Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes in the set order with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of elements to be returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see SQL Table Field Usage Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the list is the project. The context of any other attribute filter in the list is the attribute above it in the list. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the SQL Table Field Usage parameters in a separate tab, use the arrow button in the upper right corner of **SQL Table Field Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available.

Reorder current results, use this button to order the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 41. SQL Table Field Usage Tab nodes	
Node name	**Description**
SQL table	Presents a list of SQL Tables in the project. The information in the subtree, under the SQL table, lists the fields that represent where used information for the SQL table fields.
SQL table field	Presents a list of table fields.
Program	Opens the selected program at the first line.
Statement type	Statement type nodes represent the SQL statement types that are used to refer to the table. Child nodes show the name of the source file and the location (line number) of the SQL statement.

SQL Table Field Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **SQL Table Field Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **SQL Table Field Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

SQL Table Usage

The **SQL Table Usage** tab lists the database tables that are used by the project. For each table, nodes represent the different statement types that refer to the tables (select, update, delete, insert), and for each statement-type, the references in the source code (file name and location of the reference) are listed.

To display the **SQL Table Usage** tab, go to **Explore Projects** tab and select **SQL Table Usage** from the **Usage in Programs** list of analyses to the right of the screen.

The central part of the screen displays the list of attribute filters. By default, all attributes are selected; deselect the ones that you do not want to use. The breadcrumb displays the selected projects, the selected attributes in the set order with the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter that you want to use for the attributes' names. After you press **Apply**, only the elements that match the name filter are displayed in the Results area. Special characters can be used. For example:

- "*" is used to search elements, by adding to it either prefixes or suffixes as in "AAA*" or "*AAA"
- "," is used to search multiple elements as in "AAA, BBB, CCC"

Note: The special characters "*" and "," cannot be used simultaneously.

To limit the number of elements to be returned by the filtering operation, use the fields in the Limit column. The default limit is 50; the maximum limit can be set in the corresponding Usage Settings section from the Preferences dialog. For more information, see SQL Table Usage Settings. The Context column displays the context of the filter operation where relevant. The context of the first attribute filter in the

list is the project. The context of any other attribute filter in the list is the attribute above it in the list. To modify the attributes' order in the list, use the green arrow buttons to the left of the attributes list. When the attributes' order changes, the context of the filtering operation changes. To the right of Attribute filters area, the list of projects is displayed.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

By default, **Enable levels limit** is selected and the number of elements that are presented in the Results area is limited by the values set in the Limit column. If you deselect this option, the Limit column from the Attribute filters area disappears and the Results area displays all the results of the analysis. Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of Attribute filters area to hide it and maximize the Results area. To see the SQL Table Usage parameters in a separate tab, use the arrow button in the top right corner of **SQL Table Usage** tab.

The results of the filtering operation are displayed in the Results area. Use the search field to locate a specific element in the Results tree. Click the adjacent button **find only annotated** to filter the search results so that only the annotated elements are shown. In the upper right corner of the Results area, the following buttons are available:

Reorder current results, use this button to order the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

Expand all, click this button to expand the tree.

Collapse all, click this button to fold all the tree branches.

Export Results to CSV File, use this button to export the results together with the used filter configuration.

Place the mouse cursor over a node in the tree to display a tooltip that presents the element category and name. When you select a tree node, the area to the right of the screen is updated to display the graphs, reports, and usage analyses that can be generated for the selected element. Also, the **Properties** view displays the information available for the selected element.

In the tree, double-click an element to open the corresponding file at the occurrence line. The following nodes are available.

Table 42. SQL Table Usage Tab nodes

Node name	Description
SQL table	Presents a list of the SQL Tables in the project.
Program	Opens the program file at the Program ID row.
Statement type	Statement type nodes represent the SQL statement types that are used to refer to the table. Child nodes show the name of the source file and the location (line number) of the SQL statement.

SQL Table Usage statement filtering

Starting with **IBM AD V5.1.0.9**, the **SQL Table Usage** analysis has been expanded with an extra layer of filtering. It has been designed to locate more effectively specific statement types and filter them by using the new graphic user interface.

To navigate to the **Statement Type Filter** pop-up window, open the **SQL Table Usage** analysis and click on the **All** attribute present in the **Filter** column.

As a result, the **Statement Type Filter** pop-up window is displayed and contains specific statements that can be selected and filtered to any degree of granulation.

Click on **Apply/Refresh** to display only the selected statements.

Node pop-up menu

Select a node from the tree in the Results area then right-click to display the pop-up menu. The number of options available depends on the type of node selected. **View analysis source** and **Copy** options are available for all nodes. According to the selected node, several **Collapse to ... Level** options might also be available.

API Usage

The **API Usage** tab shows the API and services used in the selected project(s). To display the **API Usage** tab, go to **Explore APIs** > **select API or service** > **double click API Usage**.

The information presented in this screen is split as follows:

- the central part of the screen displays the list of attribute filters.
- the breadcrumb displays the selected projects and all attributes additional to the filters set for each attribute.

Click the **Filter** field to the right of an attribute name and enter the filter you want to use for the attributes names. After pressing **Apply**, only the elements matching the name filter are displayed in the **Results** area. Filtering can be applied by *API full name* and *Service* only.

Use **Save** to save the current filter configuration. Use **Load** to load a previously saved filter configuration.

Press **Apply** to apply the selected filters. When the filtering operation is finished, you can use the black arrow button in the upper left corner of **Attribute** filters area to hide it and maximize the **Results** area.

Note: The results of the filtering operation are displayed in the **Results** area. Use the search field to locate a specific element in the **Results** tree. In the upper right corner of the Results area, the following buttons are available:

- **Expand all**: click this button to expand all levels of the tree.
- **Collapse all**: click this button to fold all the tree branches.
- **Export Results to CSV File**: use this button to export the results and the used filter configuration.

Reorder current resultsbutton is used to order the results using a different filter configuration. A dialog window presents the list of filters that were used to perform the initial filtering and you can set them in the desired order.

API Usage pop-up menu: select a node from the tree in the **Results** area then right-click to display the pop-up menu. Depending on the selected node, several options are available.

Compare Analysis

The following analyses can be generated in this category:

- Compare Applications.
- Compare Projects in Editor.
- Compare Projects Report - for details, see Compare Projects Report in Reports chapter.
- Shared Resources Graph - for details, see Shared Resources Graph in Graphs chapter.
- Shared Resources Report - for details, see Shared Resources Report in Reports chapter.

Compare Applications

The Compare Applications analysis detects the differences between source components of the selected applications. Detected differences are sources that exist in one application but not in the other, or sources that exist in both applications but differ in content.

To compare two applications, in Application Discovery Browser perspective, from the Projects list in the central part of the **Explore Projects** tab, select the projects that contain the applications you want to compare. For details on perspectives and their components, see IBM AD Analyze Client Reference.

When you select two projects from the **Explore Projects** tab, the available graphs, reports, and usage analyses are displayed to the right of your screen.

In the Compare Analysis section, to the right of the **Explore Projects** tab double-click Compare Applications to display the Compare Action wizard.

For each selected project, the available applications are presented. Select from the lists the applications that you want to compare then click **Finish** to display the compare analysis.

When you compare two applications, the results of the structure comparison are displayed in the upper part of the screen. Navigation tools are available. You can navigate to the next or to the previous difference or change.

The results of the compare operation can also be presented in a report. Click **Generate Report** in the upper right corner of the compare results window. For details, see Compare Projects Report in Reports chapter.

Compare Projects in Editor

The **Compare Projects** analysis detects the differences between source components of the selected projects. Detected differences are sources that exist in one project but not in the other or sources that exist in both projects but differ in content.

The compare operation can be performed between two projects.

To compare two projects, in Application Discovery Browser perspective, from the Projects list in the central part of the **Explore Projects** tab, select the projects that you want to compare. For details on perspectives and their components, see IBM AD Analyze Client Reference.

When you select two or more projects from the **Explore Projects** tab, the available graphs, reports, and usage analyses are displayed to the right of your screen. When you compare two projects, the results of the structure compare are displayed in the upper part of the screen. Navigation tools are available. You can navigate to the next or to the previous difference or change.

The results of the compare operation can also be presented in a report. Click **Generate Report** in the upper right corner of the compare results window. For details, see Compare Projects Report in Reports chapter.

Tabs in IBM AD Analyze Server Manager

After you installed **IBM AD Analyze Server**, select **Start** > **Programs** > **IBM Application Discovery Analyze Server** > **IBM Application Discovery Analyze Server Manager**. The Server Manager dialog window is displayed.

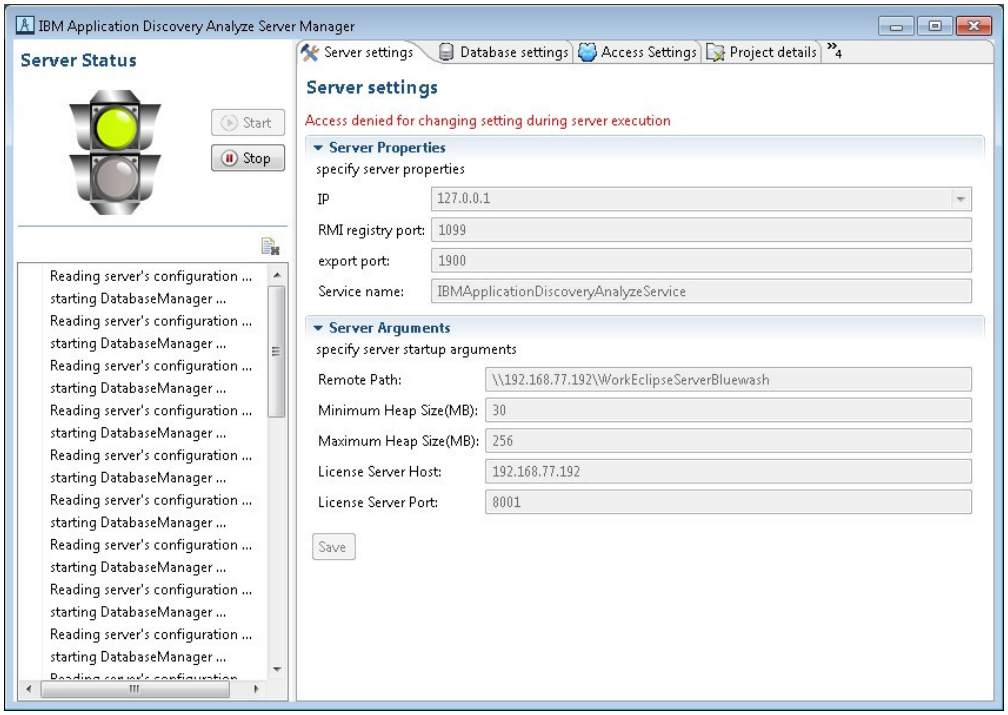


Figure 60. Server Manager Main Window

The traffic light icon in upper left corner presents the status of the server. You can use the available buttons to start or stop it. In the pane below, a list of messages details the actions currently taking place on the server. Use the button in the toolbar to **Clear status list**.

To the right of the dialog window, the following tabs are available and allow the user to set up and customize the settings of **IBM AD Analyze Server**.

- Server Settings tab
- Database Settings tab
- Access Settings tab
- Project Details tab
- Update Rights tab
- Server Requests tab
- Project Database tab

Server Settings Tab

CAUTION: The server must be stopped before any changes can be made in the current server settings.

The following Server properties are available:

- **IP**: **IBM AD Analyze Server** automatically detects all the IP addresses configured on the computer where it is installed. From the available choices, select the IP of the server that you want to use for server-client communication.

  The communication between clients and server is based on RMI. RMI requires two ports to be assigned, one for the registry and one for the communication itself. Two ports must be allocated on the server and specified both on the server side at installation time (see Server Settings in Preferences chapter) and on the client sides at server connection time.

  By default, the following ports are assigned.

  RMI registry port – 1099

Communication port – 1900

These ports must be allowed by the server firewall.

- The **Service name** field is automatically filled in by the application with the service name provided when the server was installed.

In the Server Arguments area, the following data must be entered.

- **Remote Path**, enter the correct shared network folder in the **Remote Path** field.
- **Shared folder**, for a multi-user installation, a shared folder must be created for the shared projects and *read* and *write* access must be granted to all users that operate **IBM AD Analyze**.
- **Minimum/Maximum Heap Size (MB)**, use the default values displayed in these fields or enter your own values according to optimized memory consumption.

After you selected the desired parameters, click **Save** to apply them and then start the server.

Database Settings Tab

The options available in this tab allow the user to create new databases and to modify settings for existing ones.

Important: First, verify that SQL Server authentication mode is set to SQL Server and Windows. Then, choose or add a standard user that is used to access the repository.

From the Server Settings dialog window, select the **Database Settings** tab. In the Location area, enter the following parameters.

- **Server type**, this field is filled in by default with “SQL server”.
- **Server IP**, in this field enter the IP of the computer where the server was installed.
- **Server port**, server port is the access port, usually and the default port 1433 is used.
- **Database instance**, instance name must be used in case the default database named was not chosen at SQL Server installation time.

In the Authorization area, enter the following parameters.

- **Database name**, enter a name for the database.
- **User** and **password** must match the user that was chosen in the first step.

After you entered the details of the database, click **Create database** button to create the database with the selected parameters. Use **Test database** to verify that the newly created database is functional. Click **Save** to apply the settings or changes and select the database as the working database. Use **Drop database** to delete the current database. To “select” one of the databases that were created, enter its name in the **Database name** field, and press **Test database**.

Access Settings Tab

The options available in this tab allow the user to specify allowed **IBM AD Analyze** clients, regardless of whether the server is running or stopped.

Select the **Access Settings** tab from Server Manager dialog window.

IBM AD Analyze clients can be of two types: manager and user. An **IBM AD Analyze** client with a manager role, can create shared projects, build shared projects, and delete shared projects. An **IBM AD Analyze** client with user role is a client that can import projects that are published on the server and can perform analysis. Manager and user roles are server-related attributes, which means that a server determines the type of client that is connecting to that server by looking up the client IP in the configuration files. This means that a client can be a manager on one server and a user on another server.

To add a Manager to the Managers list, click **Add** in the upper right corner of the **Access Settings** tab. The New Access Data dialog window is displayed. Enter the IP of the computer of the user who is to access the projects as a Manager (the type of owner is selected by default) then press **OK** to add the new manager to the list of Managers.

To delete one of the Managers from the list, select it then press **Delete**.

If you want to allow access to all the projects on the server to all potential users, do not add any users to the List of Users. If you want to limit the access to the projects to a number of specific users, select restrict user IP then add all of them to the List of Users. Only the users in the List of Users and List of Managers have access to the projects shared on the server.

To add a user to the list of users, click **Add** from the List of Users area of the **Access Settings** tab. The New Access Data Add User dialog window is displayed. Enter the IP of the computer of the user who is to access the projects as a User (the type of owner is selected by default) then press **OK** to add the new user to the list of Users.

Project Details Tab

This tab presents a list of projects that are published on the server and allows the user to delete a project or change the owner of any project in the list.

Note: Changes can be operated in this tab only after the server is started.

This tab presents a list of projects that are published on the server and allows the user to delete a project or change the owner of any project in the list.

To delete a project, select it from the list and press **Delete**.

To change the owner of a project, select it from the list then press **Change owner**. In the Change Owner dialog window, in the **Client settings** fields, enter the IP of the computer of the new owner and the workspace that is to be used. Alternatively you can select **Use active client** and select one of the existing clients already defined. After you defined or selected the new owner, click **Change owner**.

Update Rights Tab

This tab allows the user to select the user that has update rights for the projects that are published on the server.

Select the **Update Rights** tab from the Server Manager dialog window.

A user with update rights has access to the Update project function from **IBM AD Analyze Client**. If a user does not have update rights, the Update project function is not visible in the pop-up menu of the project.

Note that there can be only one user with update rights.

Server Requests Tab

This tab presents lists of requests that are sent by the clients to the server.

Select the **Server requests** tab from the Server Manager dialog window.

A list of requests that are sent by the clients to the server is displayed, grouped into Active, Blocked, and Not Started requests. It allows the user to select a request and then use **Stop** to display information in the output window about the selected active request, but NOT to stop processing the request. Also, information about a request from output window can be obtained by double-clicking the active request.

Project Database Tab

This tab presents a list of project databases for projects that are shared on server.

Select the **Project Database Access** tab from Server Manager dialog window.

From this tab, you can either restrict access to the databases that are public, or you can make public the databases with restricted access. To restrict access to a database, put a mark in the corresponding check-box. To make a database public, remove the check mark.

Double-click a database from the list to select it. The Projects from database and Details for database windows are populated with the corresponding data. You can view the projects that share the selected database and also the database details entered when the database was created.

Customizing Preferences

To customize the way IBM AD Analyze works, use the options available under **Application Discovery** in the **Preferences** dialog window of your Eclipse client.

To access these settings, go to IBM AD Analyze Client and then select **Window** > **Preferences** > **Application Discovery**.

In the **Preferences** dialog window, under **Application Discovery**, two main sections are available:

Environment settings
The global settings that are specified in IBM AD Configuration Server, from a web interface. These settings affect all users of IBM AD Analyze and can be modified at any time from IBM AD Configuration Server. For more information, see topic "Environment Settings" on page 429.

Local settings
The local settings that allow the user to customize the way different areas of the product (Graphs, Reports, Usage, and so on) work on the local machine. For more information, see topic "Local Settings" on page 431.

Notes:

- All the global parameters have their local counterparts. The corresponding local settings are only taken into account if the **Allow Overwrite** check box is selected in IBM AD Configuration Server.
- The available parameters in the **Preferences** dialog window depend on the installed IBM AD features.

Environment Settings

When dealing with a suite of applications that work together over a network, a problem that arises often is how to identify applications that belong to separate installations. Simply deploying every parallel installation to separate networks is not always possible, therefore, the applications from an installation must be able to locate each other and the accessible resources without interfering with applications from other installations. In the context of **IBM AD** such an installation is called "environment".

An "environment" is defined in **IBM AD Configuration Server**. For each environment a default workspace and a default blank configuration are created and then automatically attached. A workspace is a collection of projects to which the administrator can grant access to the users of **IBM AD**. A configuration holds the installation parameters for the server components and the databases that **IBM AD** uses. It also stores global parameters for Reports and Rule based reports.

In order to work correctly with the rest of the components from the same installation, **IBM AD Analyze** needs to access the environment information. It is automatically made available to **IBM AD Analyze** by **IBM AD Configuration Server**.

To access the **Environment settings**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Environment settings**.

The following information is available:

- **Host**, displays the host name or the IP address of the computer where **IBM AD Configuration Server** is installed.
- **Port**, displays the communications port for **IBM AD Configuration Server**.
- **Unique id**, displays the unique id assigned by **IBM AD Configuration Server** to the current environment.
- **Name**, displays the name of current environment, as defined in **IBM AD Configuration Server**.

Note: Make sure that the hostname or IP address, and the communications port displayed here are the ones set when you installed **IBM AD Configuration Server**. Also, make sure that the environment name and the unique ID displayed here are the ones associated to the environment you defined in **IBM AD Configuration Server**.

A list of links to the global settings for each **IBM AD** component is available. Click these links to view the global settings for each component.

Authorization and Authentication

To start working with **IBM AD Analyze Client**, make sure that all the parameters from **Environment settings** are configured. For more information, see “Environment Settings” on page 429.

If the **Authentication and Authorization** feature is *enabled*, you need to be authenticated in **IBM AD Analyze Client** to have the list of the projects displayed.

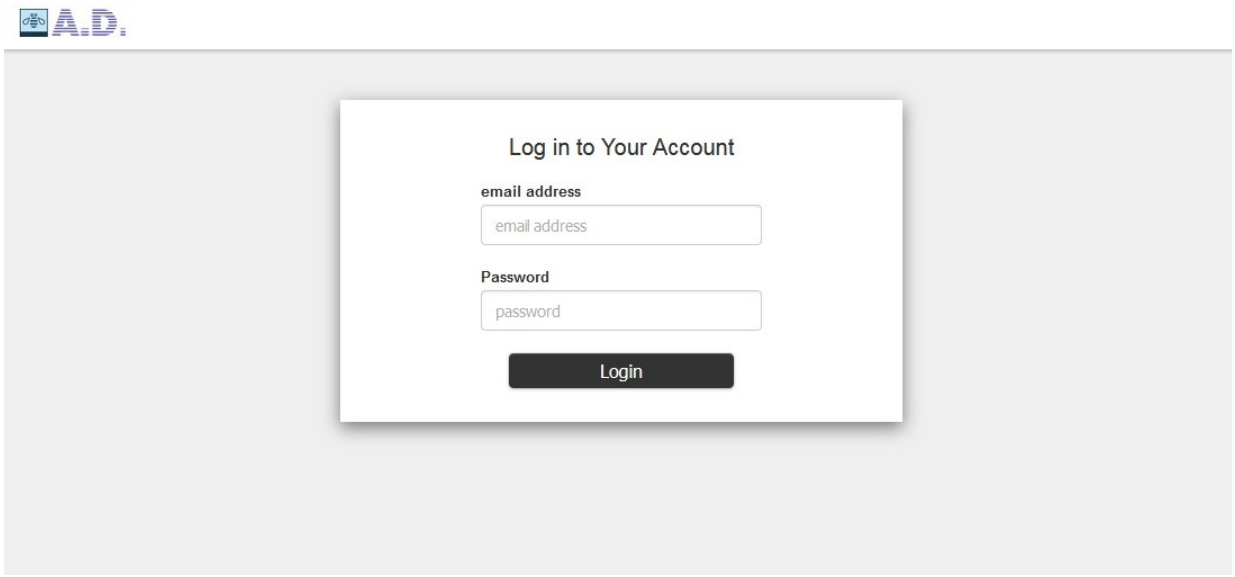
To be authenticated and authorized, follow the instructions below.

1. In your **IBM AD Analyze Client**, select the **AD Login** button.

   Note: The button is present on the right bottom corner of the screen, where the login status is also displayed.

2. The **"Log in to Your Account"** page is displayed as in the following image.

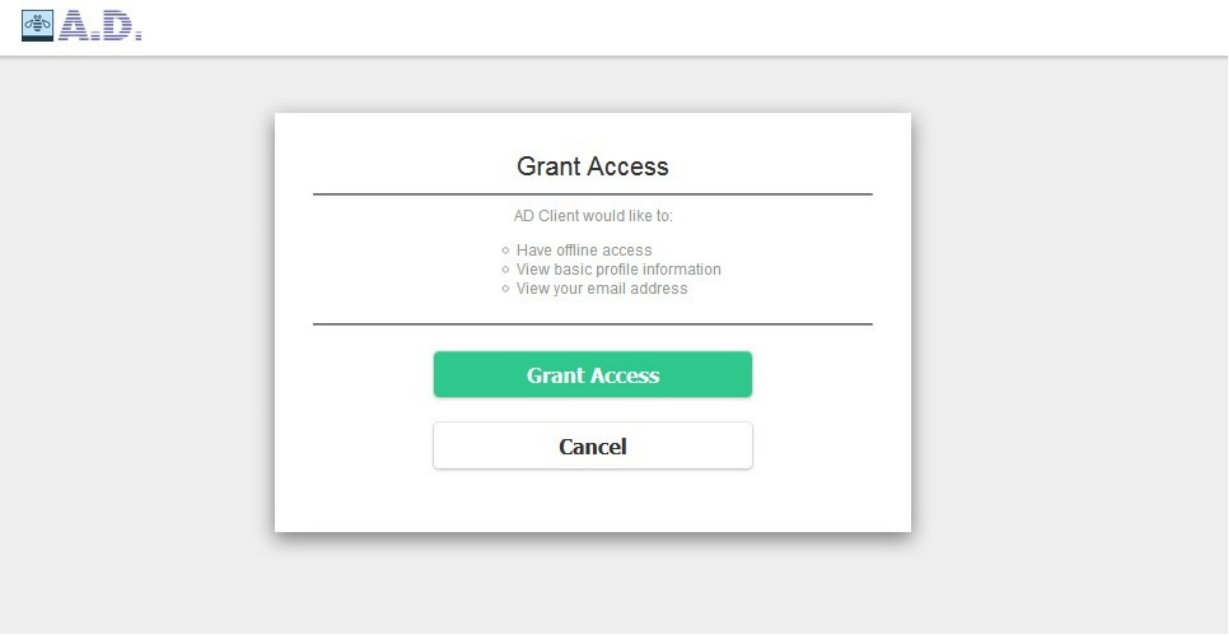
   

   Enter the following information:

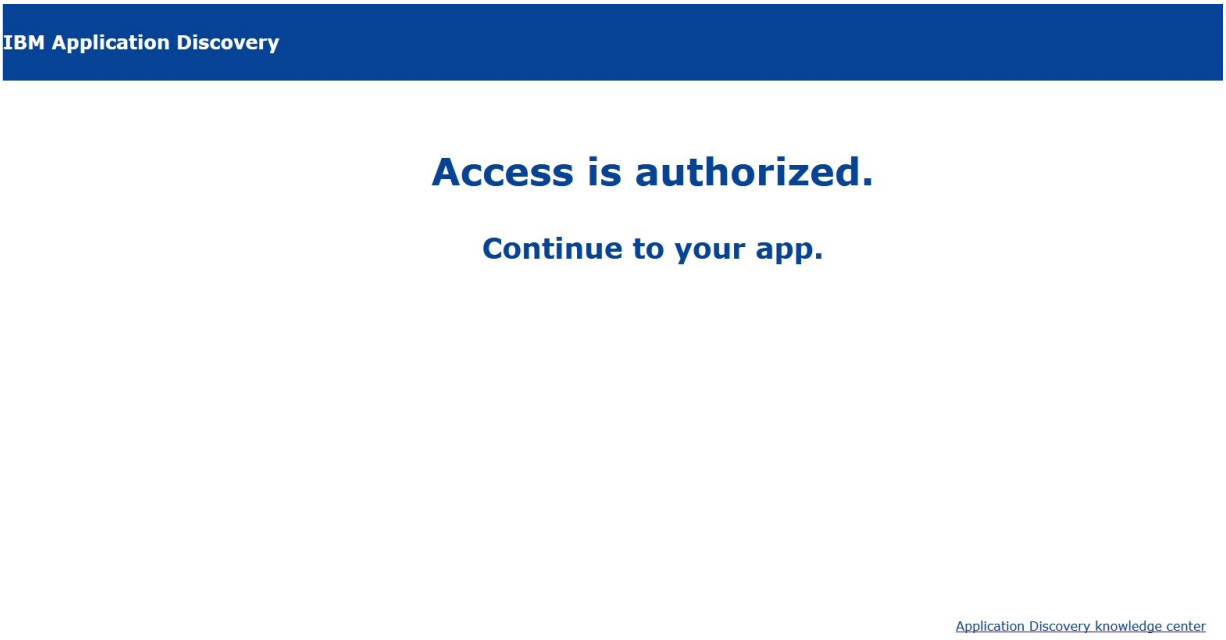
   - **Email address**: Expects the account's email address that is defined in **Secure Storage**.
   - **Password**: Expects the account's password email address that is defined in **Secure Storage**.

3. After you select **Login**, the following screen is displayed. You can choose to continue by selecting **Grant Access**.

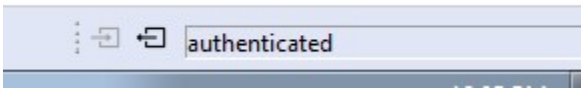
   Note: If during the configuration of **Authentication Server (DEX)**, the **skipApprovalScreen** parameter was set to *true*, the following screen is not displayed.



4. After you select **Grant Access**, the following screen is displayed, where you are informed that your access is authorized.



On the right bottom corner of the screen, the login status is displayed as **authenticated**.



Local Settings

Use the local settings to customize the way IBM AD Analyze works on the local computer. The available local settings depend on the installed IBM AD features.

Annotation Settings

Note: These settings are only relevant for Java projects.

Use **Annotations Settings** to configure the way annotations are managed in Java projects.

To access **Annotations Settings**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Annotations settings**. The following options are available, both are selected by default.

- **Automatically manage orphan annotations**, select this option to allow **IBM AD Analyze Client** to manage annotations that are no longer attached to a resource. If this option is selected, if a **Clean and Build Project** is performed, the user can recover all annotations for all project resources. If this option is disabled, if a **Clean and Build Project** is performed, annotations can no longer be recovered.

  Note: To view orphan annotations, goto **Application Discovery** and select **Annotations** > **Orphan Annotations**.
- **Show replies in annotation report**, select this check box to include in **Annotations Report** the replies that were made to annotations.

Cross Analysis

Note: This setting is only relevant for **Cross Applications Analysis** graphs, which are generated starting from Java projects.

Use **Cross Analysis preferences** to customize the way the **Cross Analysis** graphs are generated.

To access Cross Analysis options, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Cross Analysis**. The **WSBind Mapping** option is available. For more information, see "WSBind Mapping" on page 432.

WSBind Mapping

Use **WSBind Mapping** preferences to select the storage location for WSBind files.

Note: This setting is only relevant for **Cross Applications Analysis** graphs, which are generated starting from Java projects.

To access WSBind Mapping options, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Cross Analysis** > **WSBind Mapping**.

The **WSBind Mapping** option is available. A list of locations where WSBind files are stored is displayed.

Note: This setting is relevant for the cases where the cross analysis is performed by using the **Web Services** option. This setting specifies the location where WSBind files are stored.

Use **Add** to set a folder where the WSBind files are stored, use **Remove** to delete the selected location. Click **Restore defaults** to go back to the default settings. Make sure to click **Apply** after changing the parameters.

External Libraries

Use **External Libraries** to set the path to the external libraries, which are needed to generate the cross analysis.

To access the External libraries options, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **External libraries**.

In **External libraries** tab, click **Browse** to set the paths to each of the external libraries.

Explorer for z/OS

Use **Explorer for z/OS** to set the option for displaying the Application Discovery Browser perspective when an Application Discovery analysis is generated.

To access the Explore for z/OS options, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Explorer for z/OS**.

In the **Explorer for z/OS** tab, the option **Prompt** is selected by default.

General Settings

Use **General Settings** to set the connection parameters for the **IBM AD Analyze Client and Server** and **OrientDB** components. Several other parameters can also be set.

To access the general settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **General Settings**. The following general settings are available:

- **Log folder**, the default path to the log folder is displayed. Use **Browse** to set the path to the folder where the **IBM AD Analyze Client** log files are saved.
- **Temporary folder**, the default path to temporary folder is displayed. Use **Browse** to set path to the folder where the **IBM AD Analyze Client** temporary files are saved.
- **Enable easy log4j configuration syntax**, this option is selected by default and it determines Reload log configuration to use `LogConfigurator.properties` file instead of the default `log4j.properties`. When editing the default `log4j.properties` file, make sure that you use log4j syntax. For more information on `LogConfigurator.properties` and `log4j.properties` files, see Appendix 3: Log Configuration Files Examples.

Click **Reload Log Configuration** to reload the log configuration file, from where you set the level of detail you want to generate about errors.

For more information about **General Settings**, see the next paragraphs.

Catalog Settings

Use the **Catalog Settings** page to override the settings for the connection to **AD Catalog Web Service** set into CCS environment.

To access the **Catalog Settings**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **General Settings** > **Catalog Settings**.

Allow overwrite: if left unchecked, the global settings will be used. If checked, the following parameters must be manually configured:

- Host.
- Port.
- Protocol.
- Authentication.

Client Settings

Use **Client Settings** to set the IP and communications port of the computer where the **IBM AD Analyze Client** is installed.

To access the Client settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **General Settings** > **Client Settings**.

IBM AD Analyze Client automatically detects all IP addresses. From the available choices, select the IP of the client you want to use for client/server communication. Select the check-box to manually enter the client IP. If you install **IBM AD Analyze Client** on a notebook, use the local host IP address, 127.0.0.1.

Port field displays the default value for communications port.

OrientDB Settings

Use **OrientDB Settings** to view or set the details of the connection with the graph database.

Note: These settings are only relevant if an OrientDB is installed.

OrientDB or the graph database is a container of control flow information that is used for control flow analysis and the management of dynamic calls resolution. The resolution mechanism is used for showing existing dynamic calls and their solutions (whether manual or automatic).

To access the OrientDB settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **General settings** > **OrientDB settings**.

If **Allow overwrite** option is checked in **IBM AD Configuration Server** for the OrientDB (GraphDB) settings, **IBM AD Analyze** users can select **Overwrite global configuration** and proceed to change the OrientDB settings for their local workstations.

If **Allow overwrite** option is not checked in **IBM AD Configuration Server**, only the global settings are taken into account.

Test connection allows the user to test the connection with the OrientDB database.

Server Settings

Use **Server settings** to view or select the **IBM AD Analyze Server** that the current **IBM AD Analyze** Client uses. Several servers can be added.

To access the Server settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **General settings** > **Server Settings**.

By default, a list of Analyze servers is displayed as configured in **IBM AD Configuration Server** at installation time.

If **Allow overwrite** option was checked in **IBM AD Configuration Server** for the Analyze Servers settings, **IBM AD Analyze** users can select **Overwrite global configuration** and proceed to change the Analyze Servers settings for their local workstations.

If **Allow overwrite** option is not checked in **IBM AD Configuration Server**, only the Global settings are taken into account.

Test connection allows the user to test the connection with the graph database.

To select the server you want to work with, put a mark in the corresponding check-box. If you want to test the connection with any of the servers in the list, click anywhere in the associated line then press **Test connection**.

To add a server to the list, click **Add** and enter the IP of the server and the port then click **OK**.

Imports Analysis Options

Note: This setting is only relevant for Java and NetWeaver projects.

Use **Imports Analysis** options to determine which graph components are included in the **Import Analysis Graph** (for Java and NetWeaver projects).

To access the **Imports Analysis** parameters, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Imports Analysis**.

By default, all the options are selected (imported, inherited, and implemented).

License Agreement

Use **License Agreement** to read the terms of the license agreement.

To read the **License Agreement**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **License Agreement**.

Limits and Restrictions

Use **Limits and Restrictions** to limit the number of elements that are displayed in the analyses that are generated in **IBM AD Analyze Client**.

To set specific limits and restrictions for the various analyses that are generated with **IBM AD Analyze client**, go to **IBM AD Analyze client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and Restrictions**.

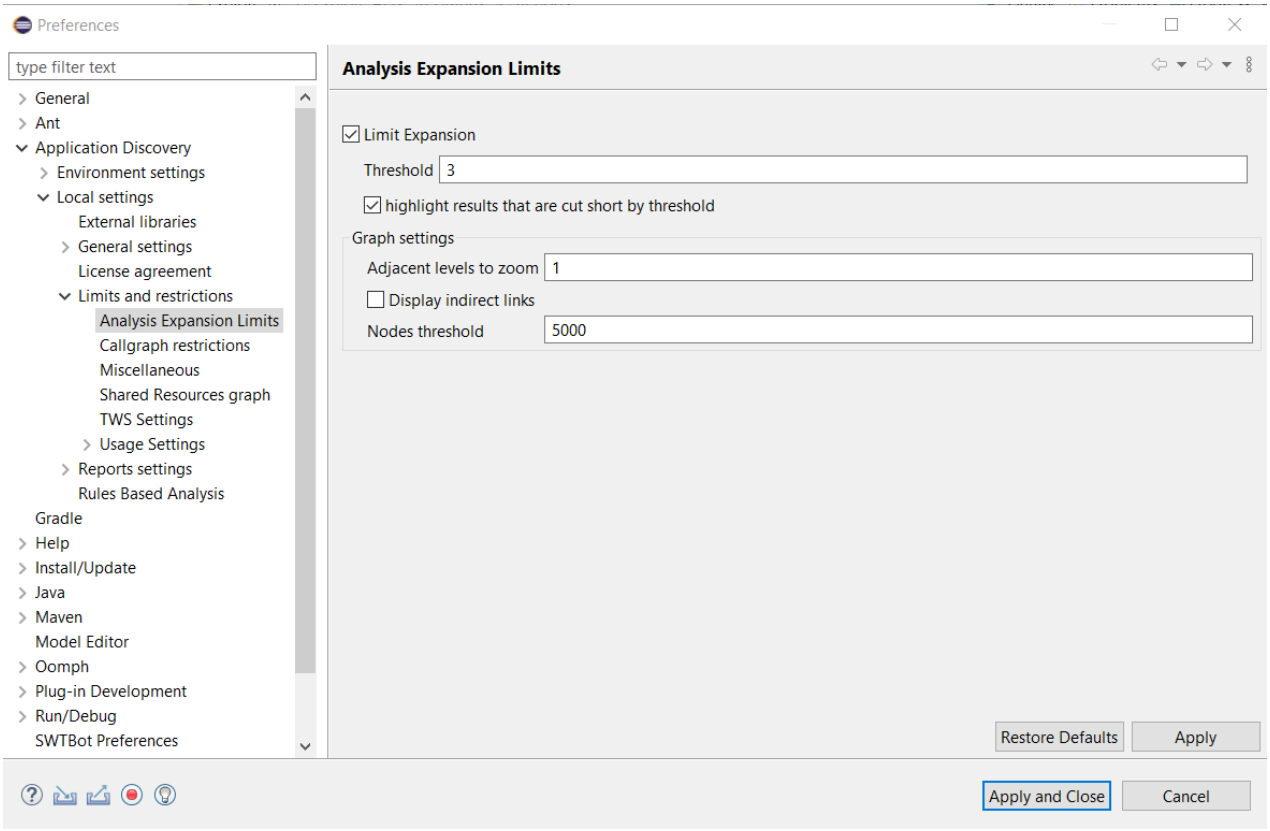
For more information about **Limits and Restrictions**, see the next paragraphs.

Analysis Expansion Limits

Use Analysis Expansion Limits to set global limits for the analyses that are generated in **IBM AD Analyze Client**.

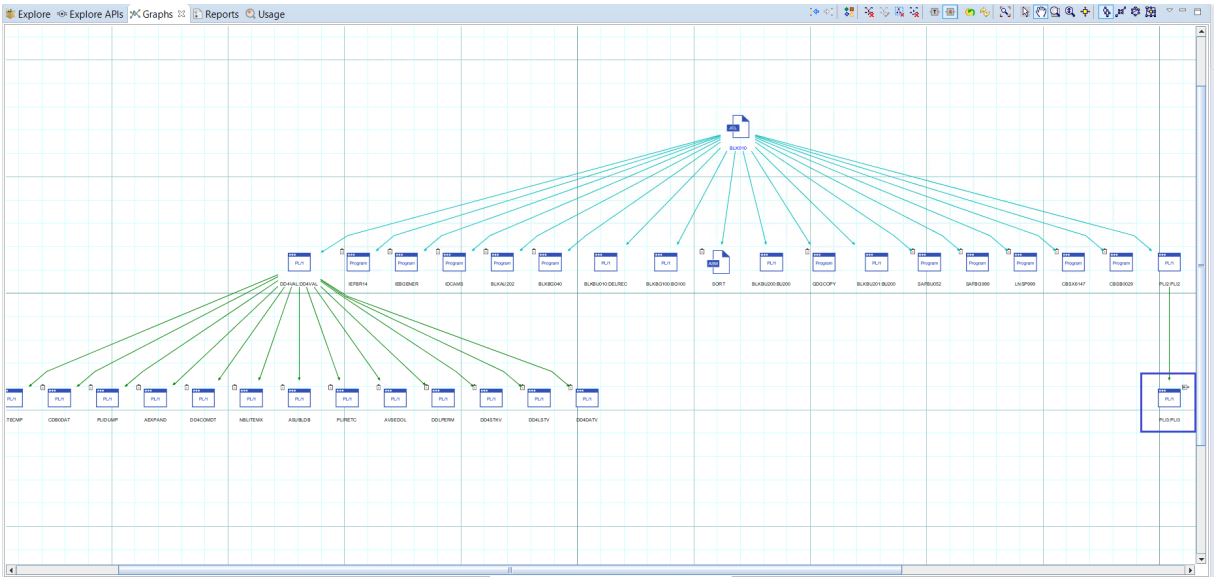
To access Analysis Expansion Limits settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Analysis Expansion Limits**.

The default configurations on the **Analysis Expansion Limits** page is shown as the following example.

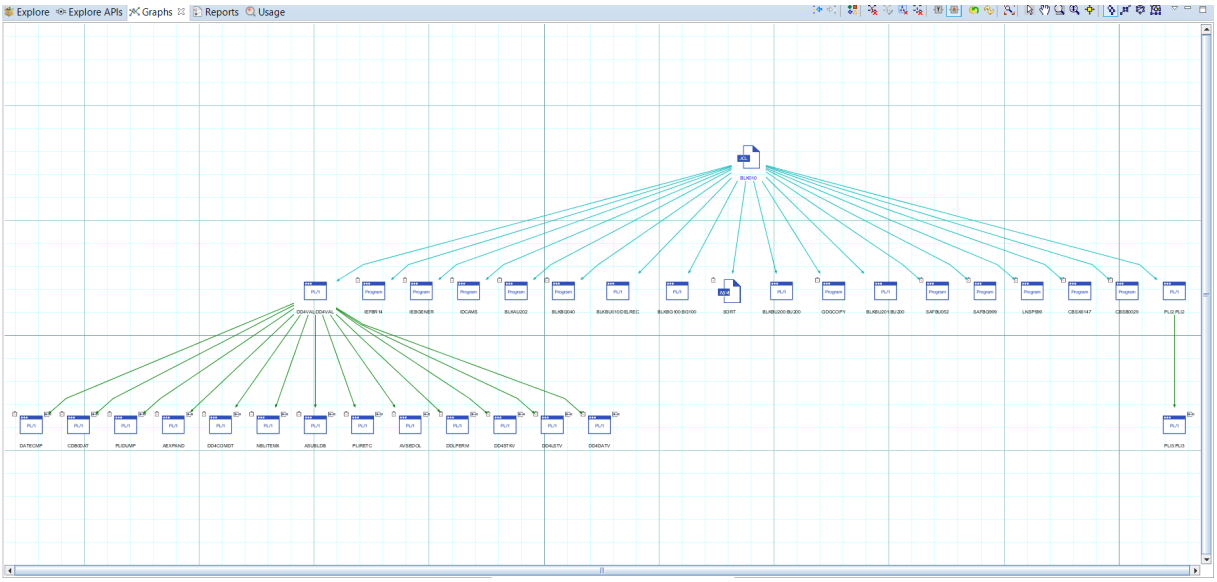


The following parameters are available:

- **Limit Expansion**, this option is selected by default and it limits the expansion of analyses that are generated with **IBM AD Analyze Client** to the number of the levels that are defined by the **Threshold** field.
- **Threshold**, enter the maximum number of levels to be processed by the analysis. The default value is "3" levels.
- **highlight results that are cut short by the threshold**, this option is selected by default and it, along with the **Threshold** value, controls how the Limit marker icon is used to highlight the nodes beyond which further analysis results are not displayed. For details, see the corresponding sections on each graph for which this option is available.
  - If the **highlight results that are cut short by the threshold** option is selected, the Limit marker icon is used to highlight only those nodes beyond which further analysis results exist and can be displayed, but which are hidden based on the value of the **Threshold** field. In this case, the Limit marker icon specifies that some information is available but intentionally not displayed. Take the following image as an example. When the **Threshold** is set to 3, and the **highlight results that are cut short by the threshold** option is selected, only **PLI3:PLI3** is marked with a Limit marker icon. This is because in the case of the threshold that is set higher than 3, or when the **Limit Expansion** option is not selected, a node that exists beyond PLI3:PLI3 called PLI4 will also be visible.



– If the **highlight results that are cut short by the threshold** option is not selected, the Limit marker icon is used to highlight all those displayed nodes that represent the defined **Threshold** value. In this situation, any node that is at the level that is defined by the **Threshold** value is highlighted with the Limit marker icon, no matter if there is one or more nodes beyond that highlighted node that are not displayed due to the value of the **Threshold** field. In this case, the Limit marker icon specifies all those nodes that represent the exact **Threshold** value. Take the following image as an example. When the **Threshold** is set to 3, and the **highlight results that are cut short by the threshold** option is not selected, all nodes on the same level with **PLI3:PLI3** are marked with a Limit marker icon, and again, no nodes beyond the threshold limit are shown.



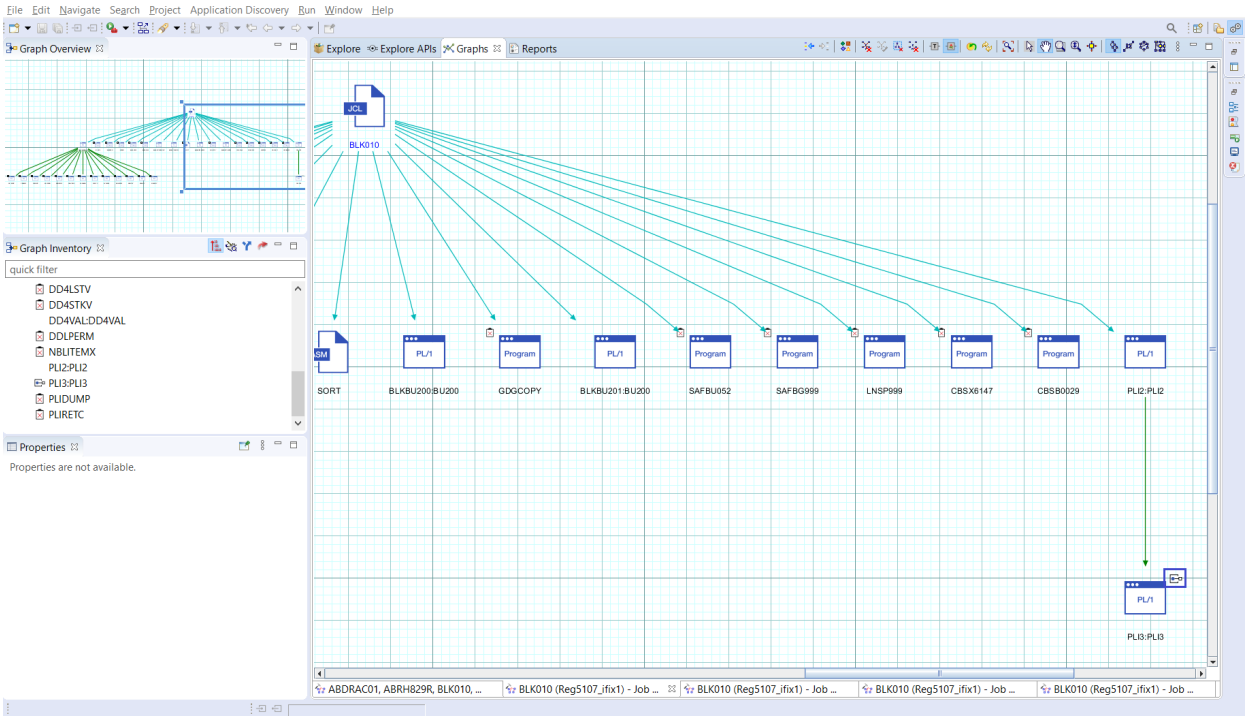
Note: This option can be set only for the following graphs: Program Call Graph, AAUTO Graph, Screen Call Graph, Job Call Graph, Transaction Call Graph, and Dataset Flow. This option can be set for the following reports: Backward Call Chains, Forward Call Chains, SQL Backward Call Chains, A-AUTO Network Dependency Report, and Data Flow Through Datasets.

- **Adjacent levels to zoom**, enter a value in this field to set the graph levels to be included in the zoom operation when a component from the graph inventory is chosen. For example, if you enter the value "3" in this field, when you double-click a component in the graph inventory, that component within the graph together with three adjacent components are displayed. Select **Display indirect links** to include the indirect links in the graph.

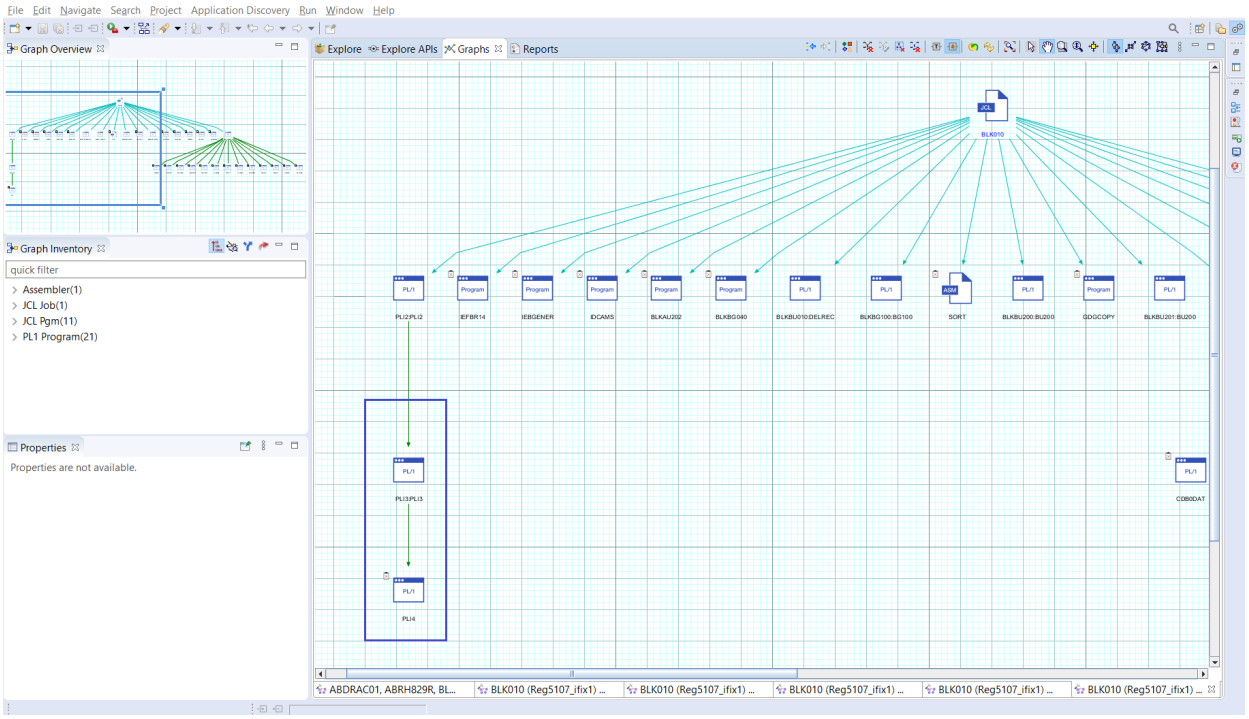
- **Nodes threshold**, enter the maximum number of nodes to be displayed in the graph. When this limit is reached, a notification appears to ask the user whether the application should display graph nodes beyond this limit or not.

Then, when you run a Job Call Graph with different scenarios, the Limit marker icon is shown in different nodes.

- If the **Limit Expansion** is selected, with **highlight results that are cut short by the threshold** selected, and the **Threshold** value is set to 3, starting from the job that is named BLK010, the Limit marker icon is shown only in the upper right corner of node **PLI3:PLI3**.



- If the **Limit Expansion** checkbox is cleared, and starts from the job that is named BLK010, no Limit Marker icon is shown in the graph because no **Threshold** value is in effect. In this case, another node **PLI4** is shown beyond node **PLI3:PLI3**.



Call Graph Restrictions

Use **Call Graph Restrictions** to restrict the number of elements that are displayed in a call graph that is generated in **IBM AD Analyze Client**.

To access **Call Graph Restrictions** settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Call Graph restrictions**.

Use the available options to specify a list of programs for which subsequent calls (the subtree below the specified programs) must not be displayed in the callgraph. If for example, you generate a forward analysis, calls starting FROM a restricted node are not be displayed but calls TO the restricted node are displayed. The same principle applies for the backward analysis. The restricted nodes appear in the callgraph clearly marked: When you work with the graph database, restricted graph nodes' names are displayed in red and also marked with a specific icon - a funnel. When you work with SQL database, restricted nodes's names are displayed in red.

Restrictions can be defined for the following call graphs: Program Call Graph, Screen Call Graph, Transaction Call Graph, and Job Call Graph.

These restrictions also apply to the following reports: Backward Call Chains, Forward Call Chains, and SQL Backward Call Chains. In the reports, a restricted node is marked with an exclamation mark, "!".

Supported patterns include *AB, AB*, *AB*.

Check the **Stop nodes will not be visible in graph** box to trigger whether the restricted nodes will be visible or not in the graph or report.

Note: If the display of further results for a graph node is limited by both a restriction set in this tab and the Nodes threshold parameter (**Call Graph limits** from the wizard), the restriction takes precedence. See the respective graphs sections for details on the icons that indicate the restrictions.

Expand Callgraphs to Other Projects

The option to expand callgraphs through APIs to resources from other projects is available for all callgraph types: API callgraphs, program callgraphs, transaction callgraphs, and screen callgraphs. To enable the expansion, click **Window > Preferences > Application Discovery > Local settings > Limits and restrictions > Callgraph restrictions** and select the **Expand Callgraph using API to other projects** check box.

- If the check box is not selected, which is the default setting, resources from other projects are displayed as nodes that are prefixed with the project names or CICS region names in the generated callgraph.
- If the check box is selected, the generated callgraph is expanded with callgraphs that start from the related resource nodes of other projects.

Note: To enable the callgraph expansion of a project, you must define the "location" element for the location of another AD project or CICS region in the resolution configuration file of the project. For more information about the file, see section JSON File Example Used for API/Macro Call Extensibility in *IBM AD Build User Guide*.

Miscellaneous

Use Miscellaneous to set various display settings for the graphs that are generated in **IBM AD Analyze Client**.

To access Miscellaneous restrictions settings, goto **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Miscellaneous**.

The following options are available:

- **Display SCL standard procedures**, select this option if you want the SCL standard procedures to be displayed in the graphs.

- **Full text for statements in Flow chart**, select this option to display the complete text of the statements in the **Flow Chart**.
- **Display temporary datasets**, select this option if you want to display the temporary data sets in the **Results** area of **Usage** tabs, in the listed data sets from **Explore** project and in the wizards for graphs and reports.

Shared Resources Graph Settings

Use Shared Resources graph settings to set the limits for the Shared Resources graph.

To access the Shared Resources graph settings, goto **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Shared Resources**.

The following option is available:

Threshold to group nodes of the same type- the default value for this parameter is "5" and it indicates that if a graph contains more than five nodes of the same resource type. These nodes are grouped in a single resource-group node. Setting up this threshold allows the graph to be kept simple and easy to understand even if many shared resources of a same type exist in the graph.

TWS Settings

Use TWS settings (Tivoli Web Services) to limit the number of elements that are displayed in the TWS graph.

To access the TWS settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **TWS settings**.

The following option is available:

Applications limit- use this field to limit the number of applications to be displayed in the TWS graph.

Usage Settings

Use **Usage Settings** to limit the number of elements that are presented in the **Results** area of **Usage** tabs in **IBM AD Analyze Client**.

The type of element to be limited depends on the type of analysis. The default maximum value for all analyses is "200", but in each **Usage** tab this limit can be set to a lower value.

To access **Usage Settings**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings**.

The following option is available:

Use levels limit, this option is selected by default and activates the limits set in each subsequent Usage Settings dialog. If this option is not selected, the number of elements that presented in the Results area of the **Usage** tab can be controlled by the values set in the Limit column in the Attributes filters area. If neither this option nor the **Enable levels limit** option in **Usage** tabs is selected, the Results area of **Usage** tabs displays the complete results of the search.

Adabas Usage Settings

Use Adabas Usage settings to limit the number of search results that are displayed in the **Adabas Usage** tab.

To access Adabas Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Adabas Usage Settings**.

In **Adabas Usage Settings** tab you can set display limits for: Adabas, Adabas Views, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Assembler Program Structure Settings

Use Assembler Program Structure settings to limit the number of search results that are displayed in the **Assembler Program Structure** tab.

To access Assembler Program Structure settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Assembler Program Structure Settings**.

In the **Assembler Program Structure Settings** tab, you can set display limits for Programs. You can use the default value or you can customize it according to the needs of your project.

Remember: Click **Apply** to save the changes.

Assembler Variable Usage Settings

Use Assembler Variable Usage settings to limit the number of search results that are displayed in the **Assembler Variable Usage** tab.

To access **Assembler Variable Usage Settings**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Assembler Variable Usage Settings**.

In **Assembler Variable Usage Settings** tab, you can set display limits for Programs. You can use the default value or you can customize it according to the needs of your project.

Remember: Click **Apply** to save the changes.

Cobol Program Structure Settings

Use Cobol Program Structure settings to limit the number of search results that are displayed in the **Cobol Program Structure** tab.

To access Cobol Program Structure settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Cobol Program Structure Settings**.

In **Cobol Program Structure Settings** tab, you can set display limits for Programs. You can use the default value or you can customize it according to the needs of your project.

Remember: Click **Apply** to save the changes.

Cobol Variable Usage Settings

Use Cobol Variable Usage settings to limit the number of search results that are displayed in the **Cobol Variable Usage** tab.

To access Cobol Variable Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Cobol Variable Usage Settings**.

In **Cobol Variable Usage Settings** tab, you can set display limits for Programs. You can use the default value or you can customize it according to the needs of your project.

Remember: Click **Apply** to save the changes.

DAL Usage in Program Settings

Use DAL Usage in Program settings to limit the number of search results that are displayed in the **DAL Usage in Program** tab.

To access DAL Usage in Program settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **DAL Usage in Program Settings**.

In **DAL Usage in Program Settings** tab, you can set display limits for: DAL, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Datacom Table Settings

Use Datacom Table settings to limit the number of search results that are displayed in the **Datacom Table Usage** tab.

To access Datacom Table Settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Datacom Table Settings**.

In **Datacom Table Settings** tab, you can set display limits for: Datacom tables, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Dataset Usage in Jobs Settings

Use Dataset Usage in Jobs settings to limit the number of search results that are displayed in the **Dataset Usage in Jobs** tab.

To access Dataset Usage in Jobs settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Dataset Usage in Jobs Settings**.

In **Dataset Usage in Jobs Settings** tab, you can set display limits for: Datasets, Jobs, Steps, and DDNames. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Dataset Usage in Programs Settings

Use Dataset Usage in Programs settings to limit the number of search results that are displayed in the **Dataset Usage in Programs** tab.

To access Dataset Usage in Programs settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Dataset Usage in Programs Settings**.

In **Dataset Usage in Programs Settings** tab, you can set display limits for: Datasets, DDNames, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

File Usage in Programs Settings

Use File Usage in Programs settings to limit the number of search results that are displayed in the **File Usage in Programs** tab.

To access File Usage in Programs settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **File Usage in Programs Settings**.

In **File Usage in Program Settings** tab, you can set display limits for: Files, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

IDMS Usage Settings

Use IDMS Usage settings to limit the number of search results that are displayed in the **IDMS Usage** tab.

To access IDMS Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **IDMS Usage Settings**.

In **IDMS Usage Settings** tab, you can set display limits for: IDMS Schemas, IDMS Sub-schemas, Records, Sets, Programs, and Statement Types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

IMS/DB Logical Usage Settings

Use IMS/DB Logical Usage settings to limit the number of search results that are displayed in the **IDS/DB Logical Usage** tab.

To access IMS/DB Logical Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **IMS/DB Logical Usage Settings**.

In **IMS/DB Logical Usage Settings** tab, you can set display limits for: Database, Segments, Statements, and Programs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

IMS/DB Structure Settings

Use IMS/DB Structure settings to limit the number of search results that are displayed in the **IMS/DB Structure Usage** tab.

To access IMS/DB Structure settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **IMS/DB Structure Settings**.

In **IMS/DB Structure Settings** tab, you can set display limits for: IMS/DB, Segments, and Fields. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

IMS/DB Usage Settings

Use IMS/DB Usage settings to limit the number of search results that are displayed in the **IMS/DB Usage** tab.

To access IMS/DB Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **IMS/DB Usage Settings**.

In **IMS/DB Usage Settings** tab, you can set display limits for the following elements: Database, Segments, Statements, and Programs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Include Usage Settings

Use Include Usage settings to limit the number of search results that are displayed in the **Include Usage** tab.

To access Include Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Include Usage Settings**.

In **Include Usage Settings** tab, you can set display limits for the following elements: Includes, Programs, and Statement Types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Job Usage Inventory Settings

Use Job Usage Inventory settings to limit the number of search results that are displayed in the **Job Usage Inventory** tab.

To access Job Usage Inventory settings, go to **IBM AD Analyze Client**, and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Job Usage Inventory Settings**.

In **Job Usage Inventory Settings** tab, you can set display limits for Jobs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

MQueue Usage Settings

Use MQueue Usage settings to limit the number of search results that are displayed in the **MQueue Usage** tab.

To access MQueue Usage settings, go to **IBM AD Analyze Client**, and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **MQueue Usage Settings**.

In **MQueue Usage Settings** tab, you can set display limits for: Queues, Queue Managers, Programs, and Statement Types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Natural Program Structure Settings

Use Natural Program Structure settings to limit the number of search results that are displayed in the **Natural Program Structure** tab.

To access Natural Program Structure settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Natural Program Structure Settings**.

In **Natural Program Structure Settings** tab, you can set display limits for Programs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Natural Variable Usage Settings

Use Natural Variable Usage settings to limit the number of search results that are displayed in the **Natural Variable Usage** tab.

To access Natural Variable Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Natural Variable Usage Settings**.

In **Natural Variable Usage Settings** tab, you can set display limits for Programs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Pgm Usage in Jobs Settings

Use Pgm Usage in Jobs settings to limit the number of search results that are displayed in the **Pgm Usage in Jobs** tab.

To access Pgm Usage in Jobs settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Pgm Usage in Jobs Settings**.

In **Pgm Usage in Jobs Settings** tab, you can set display limits for: Programs, Jobs, Steps, DDNames, and Datasets. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

PL/I Program Structure Settings

Use PL/I Program Structure settings to limit the number of search results that are displayed in the **PL/I Program Structure** tab.

To access PL/I Program Structure settings, goto **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **PL/I Program Structure Settings**.

In **PL/I Program Structure Settings** tab, you can set display limits for Programs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

PL/I Variable Usage Settings

Use PL/I Variable Usage settings to limit the number of search results that are displayed in the **PL/I Variable Usage** tab.

To access PL/I Variable Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **PL/I Variable Usage Settings**.

In **PL/I Variable Usage Settings** tab, you can set display limits for Programs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Procedure Usage in Jobs Settings

Use Procedure Usage in Jobs settings to limit the number of search results that are displayed in the **Procedure Usage in Jobs** tab.

In order to access Procedure Usage in Jobs settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **Procedure Usage in Jobs Settings**.

In **Procedure Usage in Jobs Settings** tab, you can set display limits for Procedures and Jobs. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

SQL Table Field Usage Settings

Use SQL Table Field Usage settings to limit the number of search results that are displayed in the **SQL Table Field Usage** tab.

In order to access SQL Table Field Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **SQL Table Field Usage Settings**.

In **SQL Table Field Usage Settings** tab, you can set display limits for SQL Tables, SQL Table Fields, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

SQL Table Usage Settings

Use SQL Table Usage settings to limit the number of search results that are displayed in the **SQL Table Usage** tab.

In order to access SQL Table Usage settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Limits and restrictions** > **Usage Settings** > **SQL Table Usage Settings**.

In **SQL Table Usage Settings** tab, you can set display limits for SQL Tables, Programs, and Statement types. You can use the default values or you can customize them according to the needs of your project.

Remember: Click **Apply** to save the changes.

Local Database Settings

Use **Local Database Settings** to set the local database configuration.

In order to access **Local Database Settings**, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Local Database Settings**.

In the **DB Server** area, select the type of database server you want to use: **MS SQL Server** or **HSQL**.

If you select, **HSQL** the following fields are available:

- **Server host**, enter, or update the name of the server host.
- **Server port**, enter the port number that is used for this connection.
- **Database name**, enter the database name.
- **HSQL Directory**, enter the directory in which the database is stored. You can also browse for this directory by clicking **Browse**.

If you select **MS SQL Server**, the following fields are available.

- **Server host**, enter, or update the name of the server host.
- **Server port**, enter the port number that is used for this connection.
- **User name**, enter the user name that is used to access the database.
- **Password**, enter the password that is used to access the database.
- **Instance name** enter the database instance name if exists, or leave empty.
- **Database name**, enter the database name.

Finally, click **Test connection** to verify that the database was set correctly.

Reports Settings

Use Reports settings to determine what type of information is displayed on the front page of the reports that are generated with **IBM AD Analyze Client**. Specific parameters are also available according to the project type: Java projects or mainframe projects.

In order to access Reports settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Reports settings**.

By default, the **Owner(s)**, **Organization**, **Project/Organization Role**, and **Report header image file** fields display the global settings configured for the reports in **IBM AD Configuration Server** at installation time.

If **Allow overwrite** option was checked in **IBM AD Configuration Server** for the reports settings, **IBM AD Analyze** users can select **Overwrite global configuration** and proceed to change the reports settings for their local workstations.

If **Allow overwrite** option is not checked in **IBM AD Configuration Server**, only the Global settings are taken into account.

- **Owner(s)**, enter your name or the name of the person who generates the reports.
- **Organization**, enter the name of your company.
- **Project/Organization role**, enter your position in the organization or role in the project.
- **Report header image**, use **Browse** to locate and select a header image for the generated reports.

- **Activate Application Discovery Reports view every time a report is finished**, select this check box to automatically present the **AD Reports** view when a report is generated.

Java Reports Preferences - McCabe Complexity Report

Use McCabe Complexity Report settings to set the complexity degrees used to calculate the McCabe complexity score for Java projects.

In order to access McCabe Complexity report settings for Java projects, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Reports settings** > **Java** > **McCabe Complexity**.

The following options are available:

- **Simple**, enter the complexity degree up to which a program entity is considered as simple.
- **Moderate**, enter the complexity degree up to which a program entity is considered to be of moderate complexity.
- **Complex**, enter the complexity degree above which a program entity is considered to be complex.

Use the color chip to open the **Color** window and select a color to represent this parameter in the diagrams that are presented in the **McCabe Complexity Report** for Java projects.

Mainframe Reports Preferences

Use Mainframe Reports preferences to determine what information is included by default in the reports that are generated for mainframe projects in **IBM AD Analyze Client**.

In order to access Mainframe Reports preferences, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Reports settings** > **Mainframe**.

The following options are available:

- **Show appendix**, this option is selected by default and ensures that the appendix is shown together with the report that is generated for a mainframe project.
- **Show input details in appendix**, select this option if you want to include in the appendix the input details.
- **Use current project name**, this option is selected by default and it insert the current project name on the cover page of the report.
- **Use custom name**, select this option then enter a custom name that you want displayed on the cover page of the report instead of the original project name.

Rules Based Analysis Settings

Use Rules Based Analysis settings to specify the rules based on which the Rule Based analysis is generated.

In order to access Rules Based Analysis settings, go to **IBM AD Analyze Client** and select **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Rules Based Analysis**.

By default, the **Rules file** and **File encoding** fields display the global settings configured for the Rule based analysis in **IBM AD Configuration Server** at installation time.

If **Allow overwrite** option was checked in **IBM AD Configuration Server** for the Rule based analysis settings, **IBM AD Analyze** users can select **Overwrite global configuration** and proceed to change the reports settings for their local workstations.

If **Allow overwrite** option is not checked in **IBM AD Configuration Server**, only the Global settings are taken into account.

Use **Rules file** field to locate the file that contains the rules then select the File encoding.

For an example of this type of file, see “Rules File Example for Rule Based Analysis” on page 491.

Creating a Mainframe Project

Creating a mainframe project is the first step of the process of analyzing mainframe resources.

Mainframe projects are created in **IBM AD Build**, see `IBM AD Build Client User Manual` for details on how to create a mainframe project.

Creating a Legacy AD Java Project

The first step of the process of analyzing Java resources is to create a Legacy Java project.

1. To create a Legacy Java project, go to **IBM AD Analyze Client** and select **File** > **New** > **Project**. The **New Project** window appears.

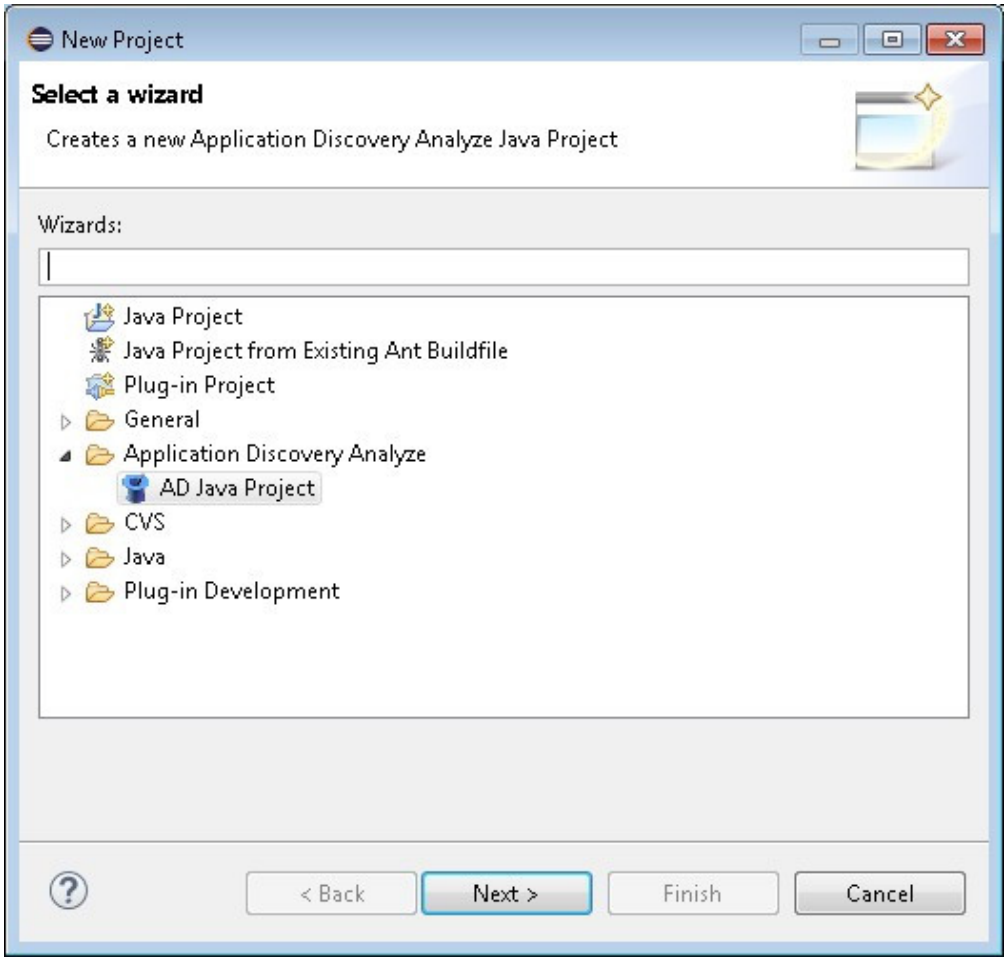


Figure 61. New Java Project wizard - project type

Under `Application Discovery Analyze` directory, select **AD Java Project** and click **Next**.

2. The **New Application Discovery Project** window is displayed.

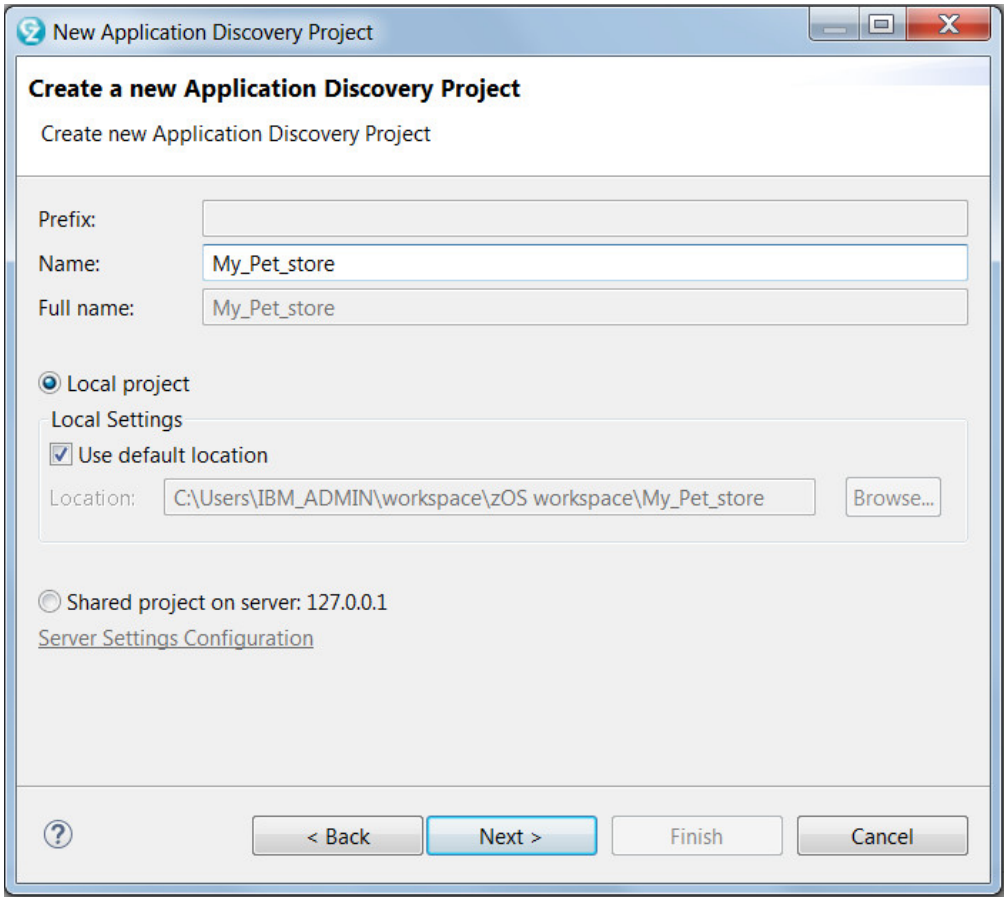


Figure 62. New Legacy Java project wizard - project title

Enter the name of the new project in the **Name** field.

Keep the project type as local project (default). By default, **Use Default Location** check box is selected. In the example presented, the project is created under the default path `C:\Users\IBM_ADMIN\workspace\zOS workspace\My_Pet_Store`. To select a different path, clear the selection of **User Default Location**: the **Location**. Click **Browse** to open the Browse for Folder window and select a location. Select **Make New Folder** to create a new folder for the new project.

In multi-user installations, projects can be shared and registered on an **IBM AD Analyze Server**. **Shared project on server...**, select this option if you want to share the newly created project. When you select this option, the associated **Server Settings Configuration** link becomes available. Click this link to open the Server settings window and select or define the server where you want to share your project. For details on how to define/change the server settings see Server Settings in Preferences chapter.

Note: In order to be able to share projects on the server, make sure that the IP of your computer is on the Allowed IPs list from the **Access Settings** tab in **IBM AD Analyze Server Manager**. For details, see Access Settings tab.

Click **Next**.

3. The Set database for project window is displayed:

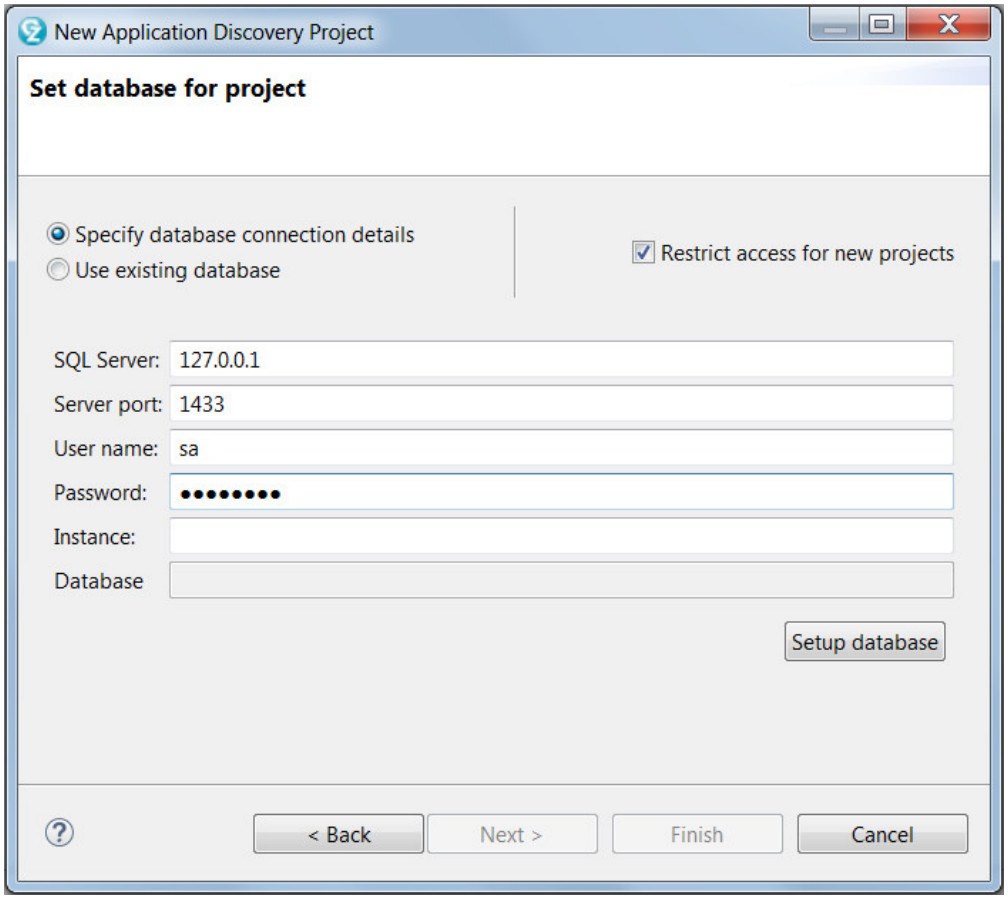


Figure 63. New Legacy Java project wizard - set project database

When new projects are created, the default option is to also create dedicated databases with restricted access. In this case, “restricted access” means that no new projects can use this database. If you want to create a database that is public, make sure that the option **Restrict access for new projects** is NOT selected.

Note: You can later change the database access parameters either from the project Properties dialog window, from the **Project database settings** tab or from the Server Manager dialog window, in the **Project Database Access** tab (for multiuser environment).

Fill in the available fields as follows.

- **SQL Server**, enter, or update the name of the server host.
- **Server port**, enter the port number used for this connection. **User name**, eEnter the user name used to access the database.
- **Password**, enter the password used to access the database.
- **Instance**, enter the database instance name if exists or leave empty.
- **Database**, this field will be automatically filled with the name of the database after it is created. This field is a non-editable field.

Once you entered the database details, click **Setup database** to create the new database for the new project. The following dialog window is displayed.

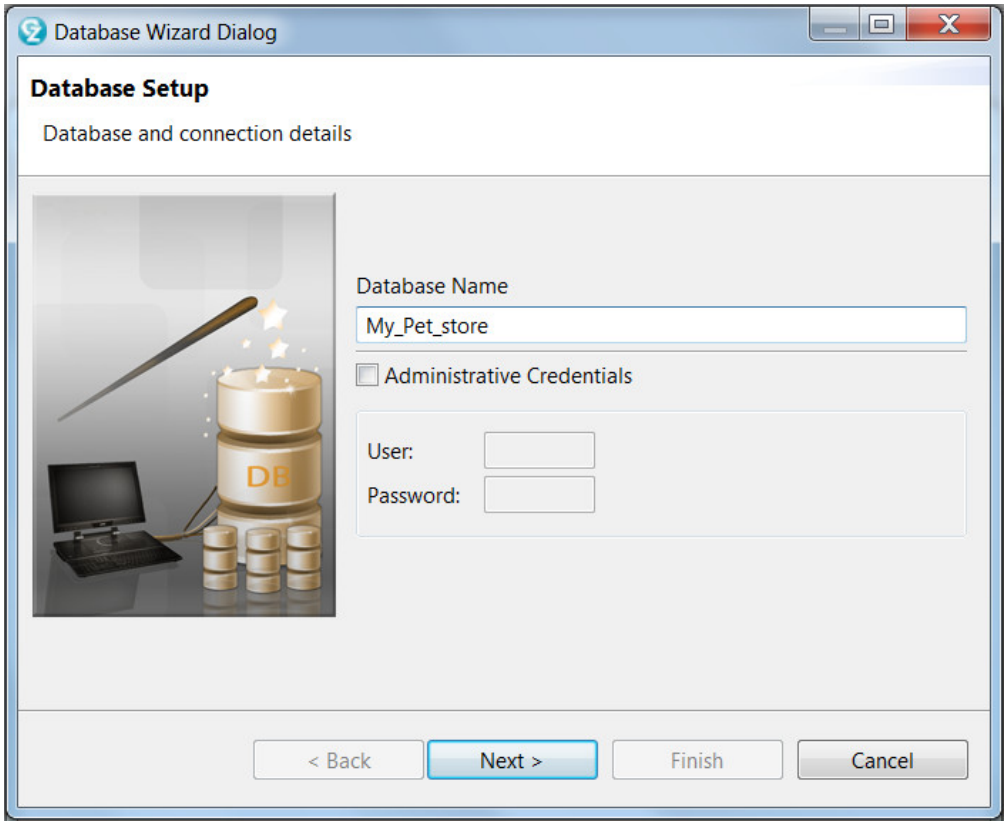


Figure 64. New Legacy Java project wizard - database setup wizard

In the first page of the wizard, enter the name of the database. When a database is configured for the first project, the wizard creates the database. If the user specified in Set database for project window does not have such an authorization, check **Administrative Credentials** and give a user and password that can be used to create the database. This user is used only for this setup procedure. Press **Next**. Five situations can exist regarding the database for the new project:

a. The database does not exist and you can create it now.

b. The database exists and is ready to be used.

c. The database exists but access to it is restricted.

d. The database exists but is obsolete.

e. The database is not valid.

Pressing the **Next** button creates or recreates the database.

If there is a connectivity issue and the database cannot be reached the possible cause is displayed on the wizard page. Fix the problem according to the possible cause, then press **Back** and try again.

Finally, after the Database setup wizard was completed successfully, the name of the database will be displayed in the **Database** field of the Set database for project wizard.

4. Alternatively, when a new project is created, instead of creating a new, dedicated database, you can use one of the existing databases. In this case you need to select the **Use existing database** option: the available options change as in the following dialog.

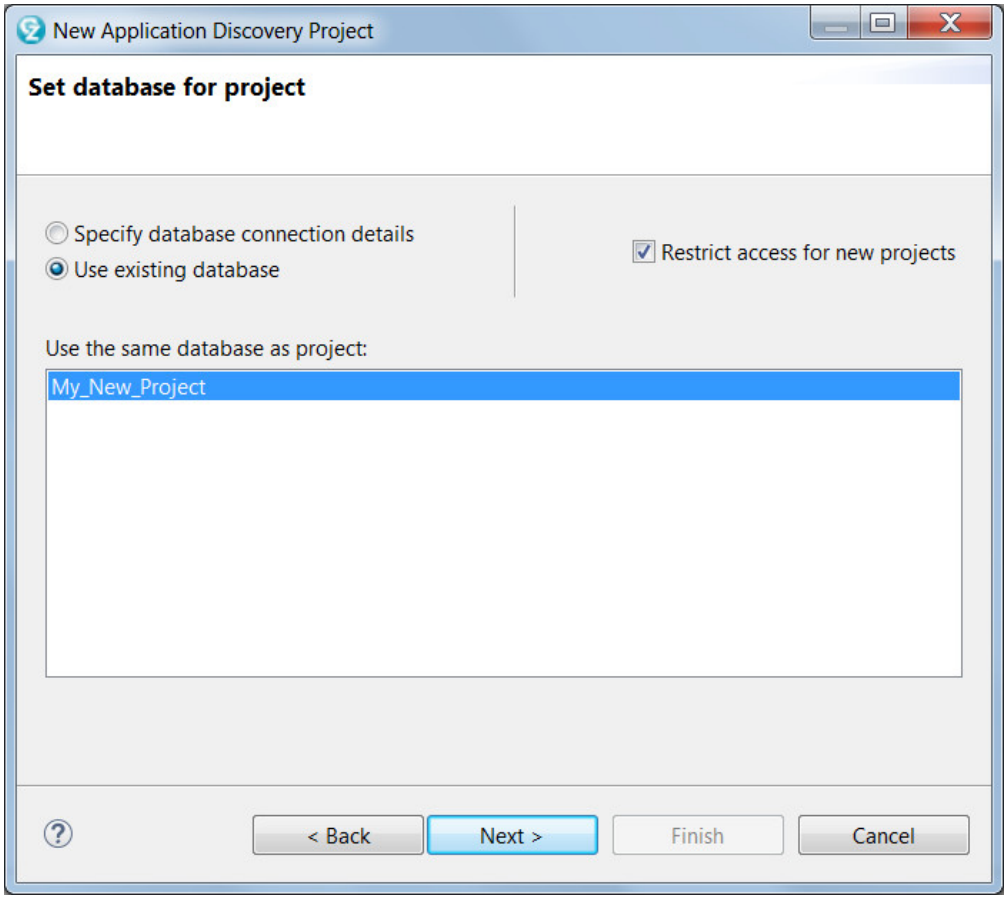


Figure 65. New Legacy Java project wizard - select existing database

The central part of the screen displays a list of existing databases that are also public. (Databases that are not public, that is, that were created with the option **Restrict access for new projects**, do not appear in this list). Select the database that you want to use for your new project. If the option **Restrict access for new projects** is selected, no other new projects are allowed to share the selected database.

Click **Next**.

5. The Project Settings screen appears allowing the user to specify the folder names for the resources to be analyzed.

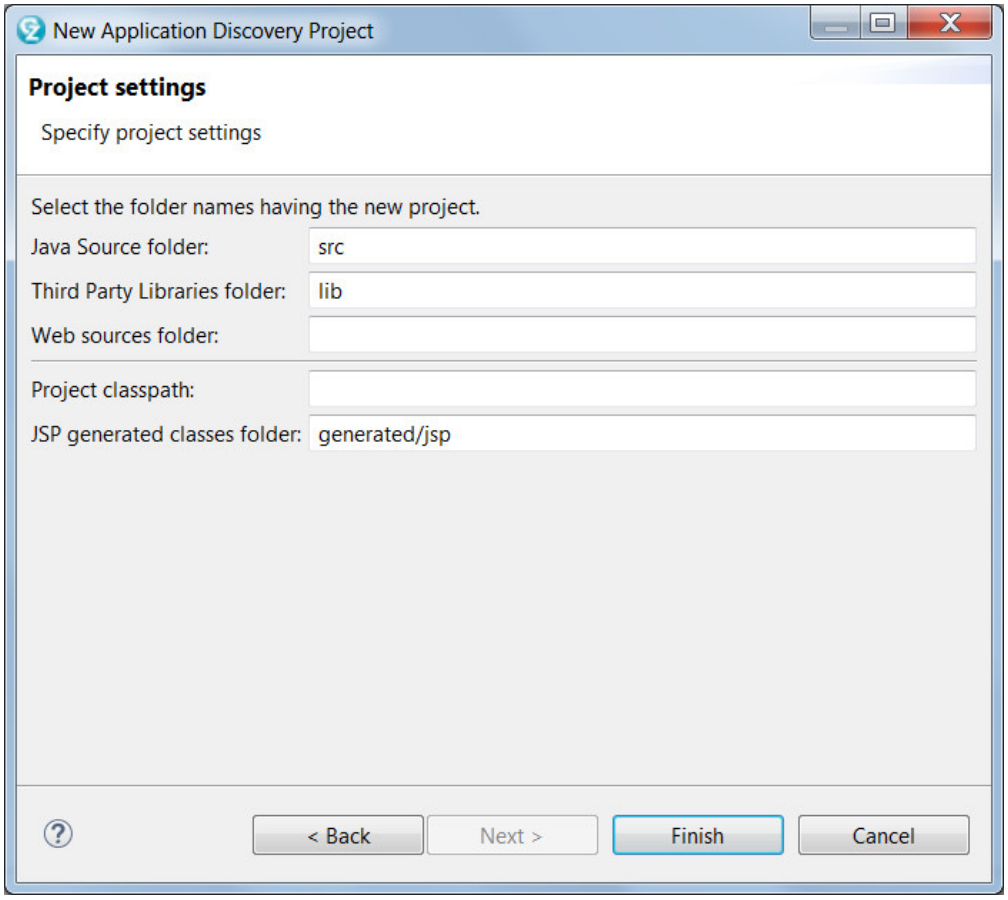


Figure 66. New Legacy Java project wizard - project settings

To accept the default directories, click **Finish**. The new project is created and is displayed as a tree structure in the **Analysis Navigator** tab on the left side of the window.

Importing a Shared Project from AD Analyze Server

Use this option to import a project that is shared on **AD Analyze Server** into **IBM AD Analyze Client**.

In multi-user installations of **IBM AD Analyze Client**, all shared projects are registered on an **IBM AD Analyze Server**. Shared projects are created by one client and then accessed by other clients for analysis purpose. In order to access such projects, the user has to import them.

1. To start an import, right-click in the AD Navigator window and choose **Import** to display the Import dialog window as in the following image.

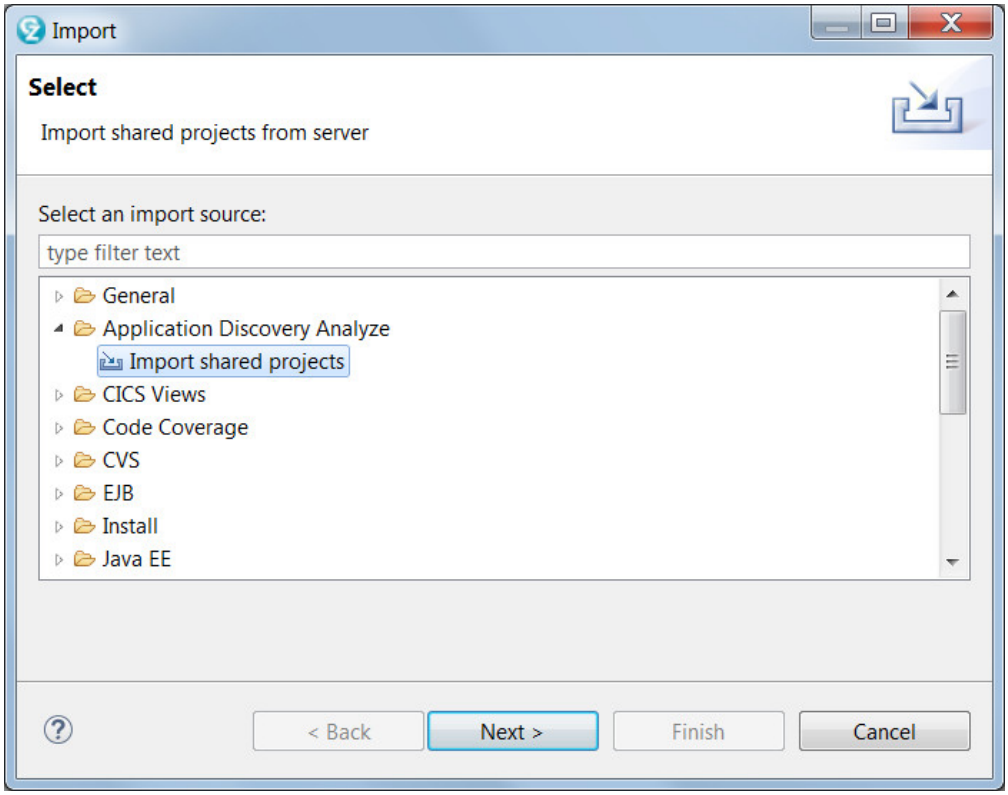


Figure 67. Import project wizard

Under **Application Discovery Analyze**, select **Import shared projects** and press **Next**.

2. The Import shared projects dialog is displayed.

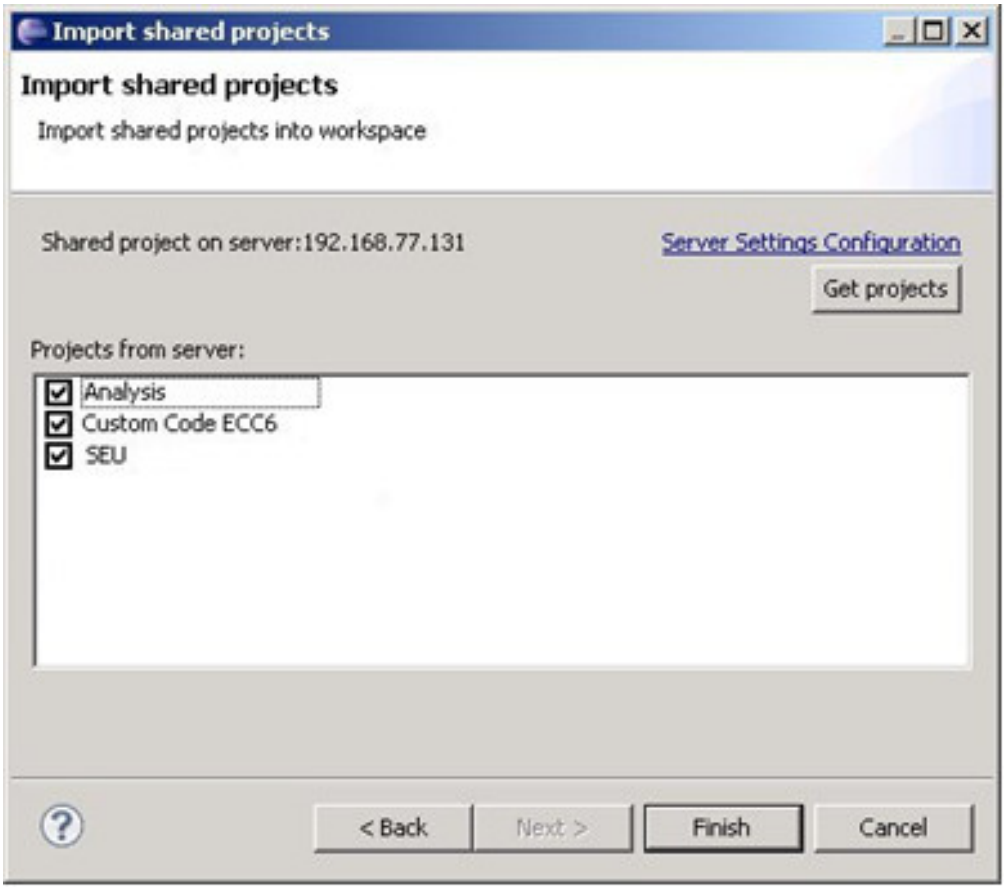


Figure 68. Import project wizard-select project to import

In the Import shared projects dialog, verify that a server name is displayed.

In case you didn't define any server yet, click the link **Server Settings Configuration**. The **Server Settings** tab from the Preferences window is displayed. Define a server, test the connection, set it as a default server, and press **OK** (for details on how to add servers, see Server Settings).

Next, press **Get projects**. In case the connection to the server can be established, and on the server there are projects that weren't imported so far to your local **IBM AD Analyze Client**, a list of projects appears. Select the project that you want to import and press **Finish**.

Checking the Connection to the Project Database

Steps that must be performed in order to check the connection to the project database.

1. To check the connection to the database, from **IBM AD Client** select **Application Discovery** > **Project Database Connection Test**.

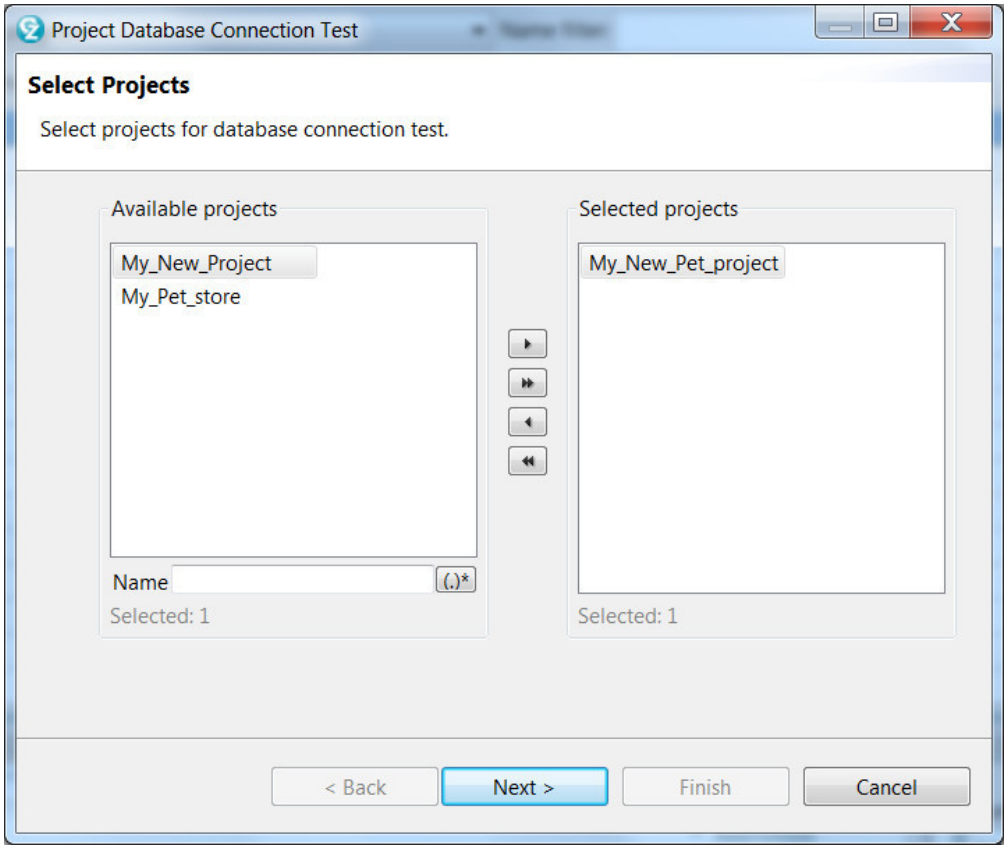


Figure 69. Select projects for which to check the connection with the database

2. The **Project Database Connection Test** wizard is displayed showing a list of projects that were identified on the server. Use the arrow buttons to move the projects from the Available list to the Selected then press **Next**.
3. The **Test Results** dialog window of the wizard presents the results of the test. For each of the selected projects the following information is displayed: project name, project database name, status of the connection, test result message text. **Project Database Connection Test** option is useful when you work with many projects: if, for example, you forgot the name of the database you associated to your project, you can use this option to identify the wanted database.

Building a Legacy AD Java Project

A “build” is the process in which **IBM AD Analyze Client** reads the project sources, places the results in the project repository, and generates the data for displaying graphical representations of the application’s internal and external program relationships.

The build process can be executed on individual programs in the project, or on all of them together. Generally, a global build is performed, but if, for example, only a single source file was changed, a build on that file alone is appropriate. In this case, only this program is analyzed and the project repository is updated. Eclipse also has an option to keep the repository up-to-date by switching on the option of an **Automatic build**. When this option is switched on, any update to the sources of the project triggers a build process to update the repository. In addition, when this option is switched on, the possibility of building the project manually is disabled. For details on the **Automatic build** option, see the IDz/Eclipse help files.

To build a project:

1. In your **IBM AD Analyze Client**, select the project that you want to build.
2. From the Project menu, select one of the Build options:
   - **Build All**, build all open projects.

- **Build Project**, build the current project.
- **Build Working Set**, build the current working set.

3. The build process starts. As each file is processed, its name and accompanying notes and messages, including error notifications, are displayed in the IBM AD Analyze Client Message Board. The **Progress** tab, and the message Sequential Build (100%) together with a Progress bar are presented in the right corner of the Status bar. The build is performed in the background. Click the **Progress** icon to show the build's progress bar. Click the **Problems** tab to view any problems that might have occurred during the build. On completion of the build, you can double-click any of these messages in order to open the corresponding source file at the appropriate line. See Problems View in Eclipse help for details on the Problems tab.
4. To update the project after several sources were changed, select the file or files to be built then right-click and select **Build Selection**.

Analyzing a Project

On completion of the build process, different functions can be performed.

- Graphically displaying the interrelations between all application entities, and/or the internal relationships within individual programs. These functions are automatically enabled after a build. For more information on the graphs which can be generated, see "Graphs" on page 330.
- Viewing and editing the application's source files. For more information on how to find a specific source file, see Explore Project.
- Producing different types of reports to analyze the data. For details, see "Reports" on page 89.
- Searching the source files for specific text strings. See Search in Files tab for details.

Handling Annotations

Annotations Overview

Annotations are comments that help the next person in the application's lifecycle to process the application. In this way, the engineers who access the annotated resource can see what additions/updates were made during the application's lifecycle and can add their own comments. **IBM AD Analyze Client** provides simple tools to assist you in handling the annotations in your project. Changes made to annotations are stored for maintenance under numbered revisions. An annotation that is created on a resource in a project is visible on all projects that contains that resource.

In case you are not using the API feature, annotations can be added and seen only on physical resources such as programs and JCL Jobs.

In the context of API feature, the annotations are collected by the API Resolver and can be seen on the following resources: `CICSTran`, `SQLTable`, `SQLField`, `CICSMap`, `IMSMessageDescriptor`, `IMSTransaction` and `IMSDBSegment`.

The options for handling annotations are available in **IBM AD Analyze Client** in multiple locations:

- In **Explore** tab, under Explore projects view, the area to the right of the screen presents the annotations option available for the selected project or projects. The option is **View Annotation**.
- In **Explore** tab, under Explore project view, when any program or JCL is selected. The options are **Add Annotation** and **View Annotations**.
- In **Explore** tab, under Search in files view, for the selected project or projects. The option is **View Annotations**.
- In the pop-up menu of the graph elements, when any program or JCL job nodes are selected from **Graph Inventory** or from the graph displayed. The options that appear in the pop-up menu under Annotations are **Add Annotation** and **View Annotations**.
- In **Usage** tab, when any program or JCL is selected. The options are **Add Annotation** and **View Annotations**.

- In **Cobol Source analysis** view, when a Cobol program is selected. The options are **Add Annotation** and **View Annotations**.

The annotation-related options are described under "Adding an Annotation" on page 457 and "View Annotations" on page 458 chapters.

Note: Make sure that Annotations Database configurations from **IBM AD Configuration Server** are set and **IBM AD Batch Server** is configured to access **IBM AD Configuration Server** that contains the configurations for annotations database. For more information, see "Server Properties" on page 78.

Adding an Annotation

Add Annotation allows the user to create a new annotation for the selected resource. An annotation can be added from Explore project view, a graph, an usage or Cobol Source analysis view.

This task takes you step-by-step through the process of adding an annotation.

1. To add an annotation, select a resource and depending on the analysis from which you want to add the annotation, you will either have to:
   a. Go in the results area and choose **Add Annotation** from the **Explore** area to the right of the tab.
   b. Go in the Graph inventory or on the graph that is displayed, right-click on the selected resource and from the pop-up menu that is displayed select **Annotations** > **Add Annotation**. After **Add Annotation** is selected the **Create Annotation** window appears.

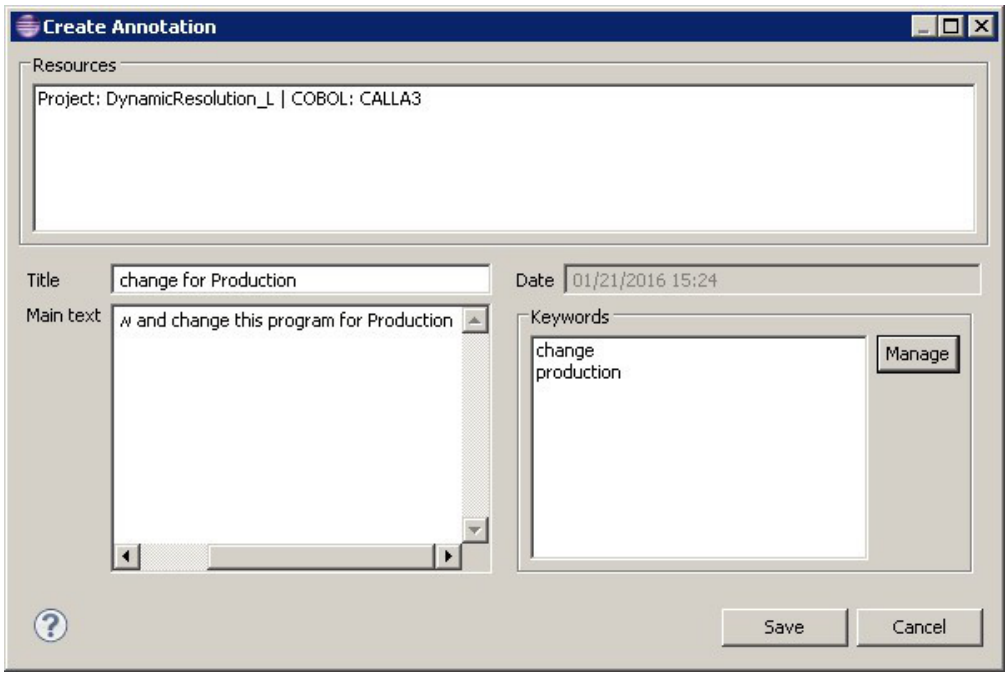


Figure 70. Add Annotation

2. The Resources area displays the resource name, the resource type, and the name of the project.
   a) In the **Title** field, enter a short, relevant title for the annotation.
   b) In the **Main text** field, enter the annotation.
   c) The **Date** field displays the current date, which represents the annotation date.
   d) To add keywords for the annotation, click the **Manage** button from the Keywords area. In the **Manage keywords** dialog window, a list of available keywords is displayed. Use the arrow buttons to move keywords between the Available and Selected areas. To add a keyword to the list, enter the word in the field at the top of the list then press the **Add** button. Once you selected all the wanted keywords, press **Ok** to return to the **Create Annotations** dialog window. The selected keywords are displayed in the Keywords area of the dialog window.

3. Finally, click **Save** to add the annotation to the selected resource. An icon next to the resource in the graph indicates that an annotation is available for that resource.

View Annotations

View Annotations presents a list of annotations that are defined for the selected resource either by a user directly in IBM AD Analyze Client or collected by API Resolver.

This analysis can be started either at resource level by selecting the resource that has attached an annotation or at project or project level by selecting one or more projects.

To add an annotation, select a resource and depending on the analysis from which you want to add the annotation, you will either have to:

1. Go in the results area and choose **View Annotation** from the **Explore** area to the right of the tab.
2. Go in the Graph inventory or on the graph that is displayed, right-click on the selected resource and from the pop-up menu that is displayed select **Annotations**, then select **View Annotation**.

To view the annotations that are attached to the resources of a project, select the project or the projects for which you want to see the annotations and from the area to the right of the screen choose **View Annotations**. The **View Annotations** tab is presented as in the following image.

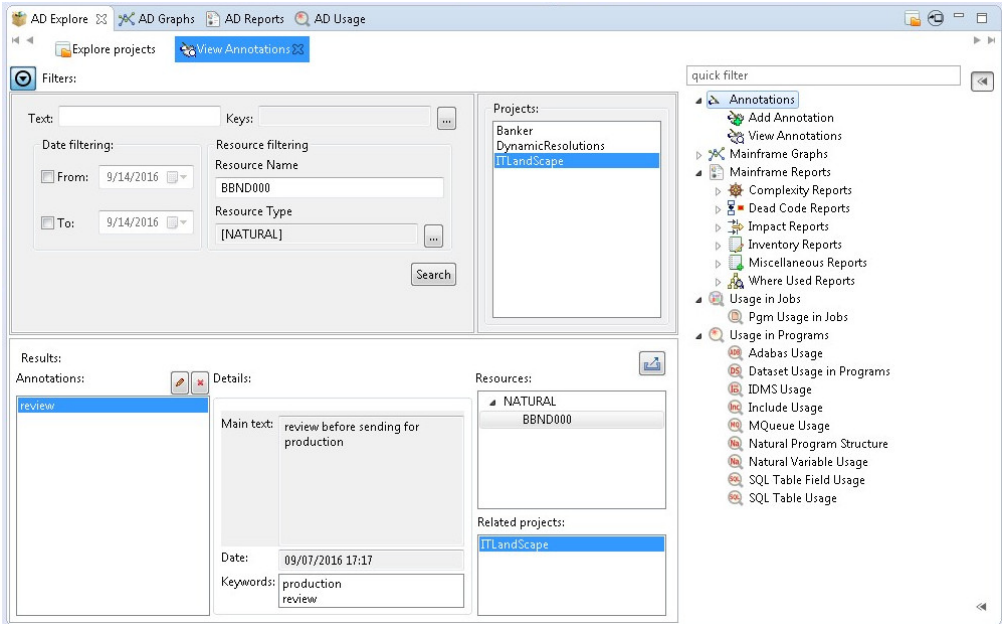


Figure 71. View Annotations

The Filters area allows the user to search throughout the selected project for annotations that match specific search criteria such as:

- **Text**, use this area to search by both annotation title and main annotation text.
- **Keys**, click the adjacent button to display a list of keywords that you can use to search for the annotation.
- **Date filtering**, use these fields to filter the search results by the date when the annotation was added.
- **Resource name**, enter the name of the file for which you want to view annotations; alternatively it shows the name of the selected resource.
- **Resource type**, click the adjacent button to display a list of resource types and select the ones that you want to use for the filtering operation.

Once you selected the wanted criteria, click **Search** to perform the actual search operation.

The Projects list area to the right of the Filters area displays the projects list and allows the user to search through one or several projects.

When the filtering operation is finished, you can use the black arrow button in the upper left corner of the **View Annotations** tab to hide the filters and maximize the Results area.

In the Results area to the left, the list of annotations is presented. You can move through the list of annotations by using the keyboard arrow keys or by selecting one from the list. If you select an annotation from the list, the fields in the Details area display the main annotation text, the date when the annotation was created, and the corresponding keywords.

The Resources area displays the resources where this annotation was found, grouped by resource type, while the Related projects area displays the list of projects that contain the resources in the above field.

Use **Edit** and **Delete** at the top of annotations list, to edit or delete the selected annotation.

Note: In case that API feature is used, only the Delete option is available on an annotation.

To export the annotations list, use **Export results to CSV file** in the upper corner of the Results area.

By selecting the resource that is displayed in the Resources area from the Results that are displayed, according to the resource type, a list of analysis are shown in the Explore area to the right of the Explore tab.

Chapter 9. Extended Capabilities

Activating the full functionality of IBM AD from a free version

A free **IBM AD** version is offered for evaluation purposes. This *for evaluation* version allows you to create a maximum of 5 projects and compile a maximum of 100 programs and 100 JCLs. After you purchase the full version, you will receive an activation tool. Following are the steps you need to take for unlocking the full functionality of IBM AD.

Stop **IBM AD Build Client** and **IBM AD Administration Tool** before using the activation tool and make sure you run it on all machines where **IBM AD Build Client** is installed.

1. Double-click the `ADActivation.exe` received from IBM. (Highly recommended to use *Run as Administrator* method)
2. **Application Discovery License Activation** dialog window is displayed. Click **Activate** then click **Exit** to finalize the activation.

Integrating with IBM License Metric Tool

IBM AD generates IBM Software License Metric Tag (SLMT) files. The versions of IBM License Metric Tool that support IBM Software License Metric Tag can generate License Consumption Reports.

Each instance of AD generates an IBM Software License Metric Tag file. The **USER** metrics are monitored. The values are refreshed every 15 minutes.

IBM AD generates an SLMT tag file that records the active user count at 15-minute intervals. The recorded information is:

- **AUTHORIZED_USER**: containing the number of the authorized active account that used the IBM AD Analyze Client during a 15-minute period.
- **FLOATING_USER**: containing the number of the unauthorized active account that used the IBM AD Analyze Client during a 15-minute period.

You can find the IBM SLM tag files under the IBM ADDI installation directory.

Following is an example of the content of an SLM tag file:

```
<SchemaVersion>2.1.1</SchemaVersion>
    <SoftwareIdentity>
        <PersistentId>4cb9f56ab26c41cc8b16670b3d87636a</PersistentId>
        <Name>IBM Application Discovery</Name>
        <InstanceId>AD/.</InstanceId>
    </SoftwareIdentity>
    <Metric logTime="2020-04-30T22:33:20+00:00">
        <Type>AUTHORIZED_USER</Type>
        <SubType></SubType>
        <Value>4</Value>
        <Period>
            <StartTime>2020-04-29T06:59:11+00:00</StartTime>
            <EndTime>2020-04-29T07:14:11+00:00</EndTime>
        </Period>
    </Metric>
    <Metric logTime="2020-04-30T22:33:20+00:00">
        <Type>FLOATING_USER</Type>
        <SubType></SubType>
        <Value>1</Value>
        <Period>
            <StartTime>2020-04-29T06:59:11+00:00</StartTime>
            <EndTime>2020-04-29T07:14:11+00:00</EndTime>
        </Period>
    </Metric>
```

© Copyright IBM Corp. 2010, 2022

Besides SLMT, AD provides another way to manage license usages through the IBM AD Audit service web GUI. For more information, see "Audit Service Web GUI Overview" on page 50.

Integrating IBM AD Analyze Client with IDz features

When **IBM AD Analyze Client** is installed together with **IDz**, a contextual menu entry (**Application Discovery** > **Analyze**) is available in **IDz** views, allowing the user to run AD analyzes.

The contextual menu is available for the following types of resources:

1. From the **Remote Systems** view:
   - PDS
   - PDS Member
   - Sequential Data Set
   - VSAM Data Set
2. From the **Local Transactions** view:
   - CICS Transactions
3. From the **Data Source Explorer** view:
   - SQL tables
4. From various **CICS Interdependency Analyzer (IA)** views, for example **Programs** or **Command flow**:
   - Programs

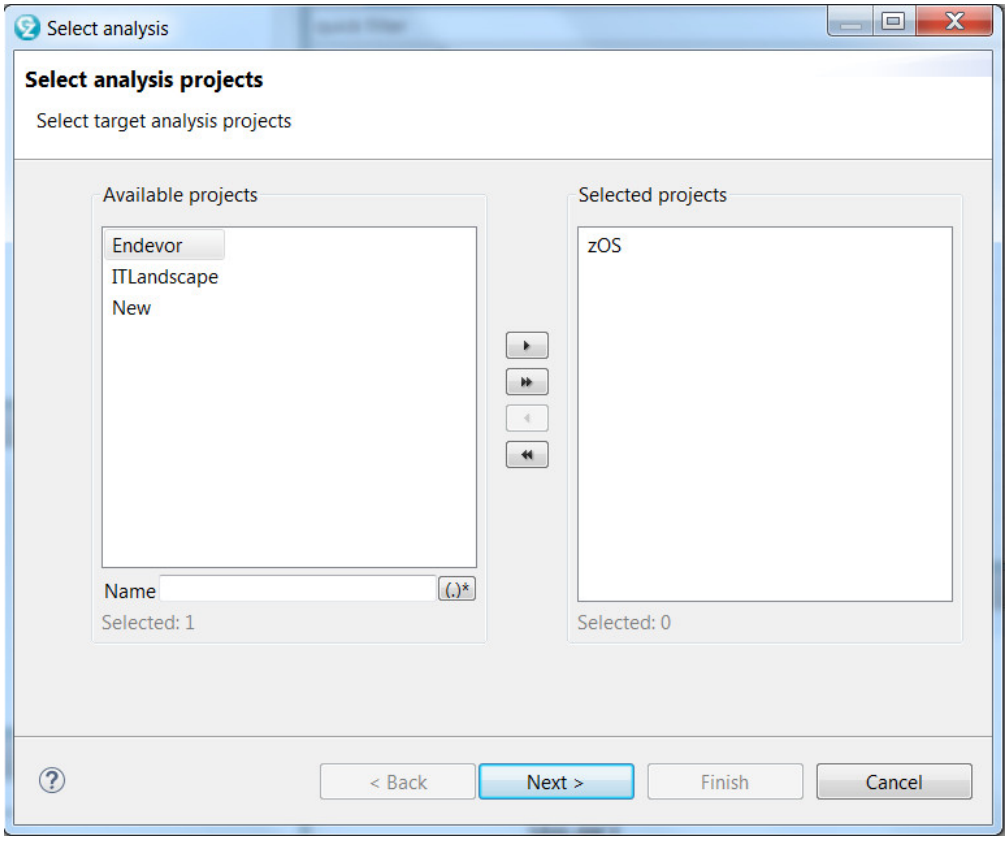
Once the action is selected, you are prompted to choose the **AD Project** that is used for analysis. Click **Next** and choose the analysis that you want to perform. Once you click **Finish** the results are displayed in AD perspective.

Analyzing Resources From a Remote Mainframe System

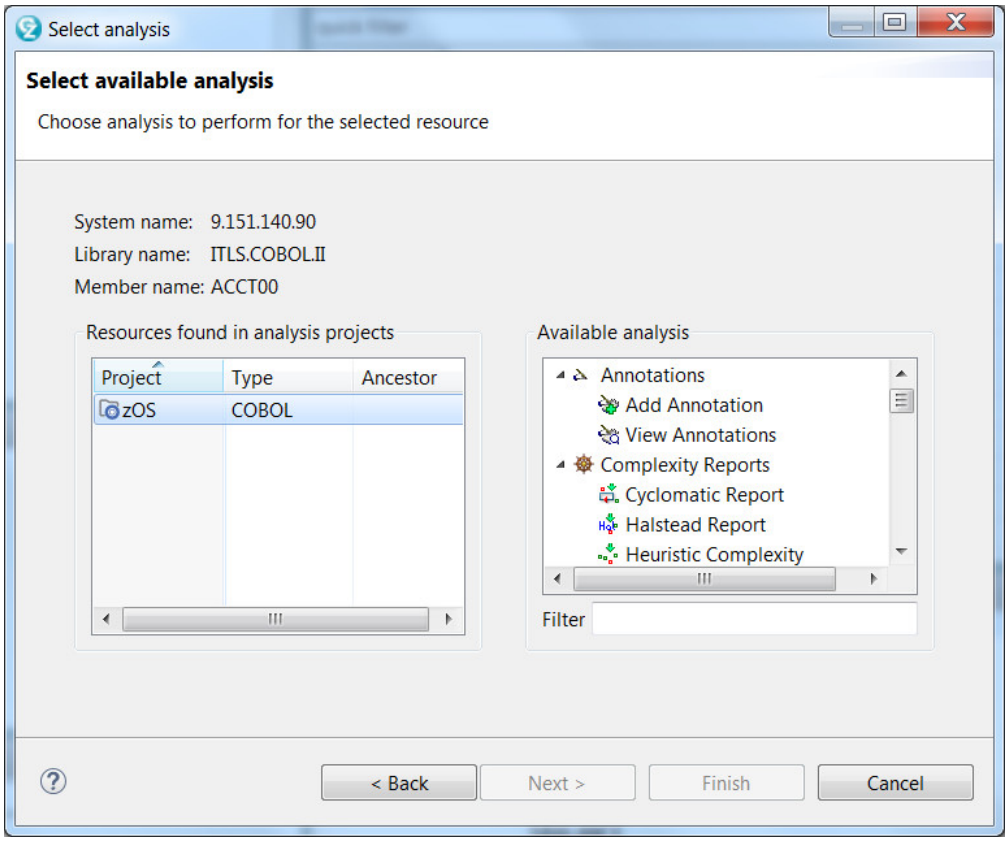
To start an analysis on a resource from a remote mainframe system follow these steps:

1. Create a project in **IBM AD Build** and bring resources from the mainframe system:
   a. In **IBM AD Build Client**: Create a project.
   b. In **IBM AD Build Configuration**:
      - Create a z/OS connection.
      - Associate the z/OS connection to your project.
      - Configure the z/OS connection.
   c. In **IBM AD Build Client**:
      - Bring resources from the remote mainframe system by using the defined z/OS connection.
      - Build your project. For details on how to create a project in **IBM AD Build**, and how to bring resources from mainframe systems, see `IBM AD Build Client and Server User Guide`.

      Once you completed these steps, when you open your **IBM AD Analyze Client** the new project is available in the projects list in the **Explore projects** tab of **AD Explore** view (to make sure that the project list is up-to-date, right-click in the projects list and select **Get project list**).
2. In your **IBM AD Analyze Client**, define a connection to the remote mainframe system. For details on how to create a connection to a remote system, see the `IDz` documentation.
3. Once the connection to the remote mainframe system was established, and you can see it in the **Remote Systems** view, set up a filter to make it easier to locate your sources. For details on how to create a filter, see the `IDz` documentation.
4. In the corresponding view from IDz, right-click the resource that you want to analyze, and from the pop-up menu select the **Application Discovery > Analyze**. The wizard dialog is displayed showing a list of projects that are found on **IBM AD Analyze Server**.



5. Select the mainframe project that contains the resource you want to analyze and click **Next**.
6. The **Select** available analysis dialog window is displayed.



The analyzes that can be performed on the resource that is selected from the remote system are displayed.

From the analysis that is generated in this way, you have the option to view the contents of a selected resource directly on the remote mainframe system, by selecting an element from the graph, and then **right click** > **View analysis source**. The contents of the selected resource are displayed in the **Editor** view and the tooltip displays the full path to the resource location on the mainframe.

z/OS Subsystem and Third-Party Product Configuration Checklists

The purpose of this appendix is to provide checklists for configuring IBM Application Discovery to interact with common z/OS subsystems and third-party tools. These configurations require changes both to IBM AD Build Configuration and IBM AD Connect for Mainframe on z/OS.

Db2 Checklist

The following steps must be performed to enable IBM AD to interact with Db2 on z/OS:

1. Add the primary Db2 installation load library to the z/OS Connector/Listener procedure JCL. For more information, see "Configuring the Listener PROC" on page 152.
2. Customize and submit the Db2 bind jobs.

   Note: You can create a DBRM-based plan, or a more current package-based plan (recommended).
3. Configure the IBM AD Build Server to access Db2 (Db2 subsystem name/level of Db2).
4. Manually access Db2 information through IBM AD Build Configuration.

For detailed information about the above-mentioned steps, check the **IBM Application Discovery Connect for Mainframe** documentation.

ChangeMan® ZMF Checklist

The following steps must be performed to configure ChangeMan® ZMF:

1. Modify the sample JCL member **IAYXMLRQ**.
2. Allocate the **XMLIN** data sets.
3. Allocate the **XMLOUT** data sets.
4. Additionally, set up **Continuous Rule Validation**.
5. Follow the steps from "Bringing Data From Mainframe Using ChangeMan® ZMF" on page 235.

For detailed information about the above-mentioned steps, check the **IBM Application Discovery Connect for Mainframe** documentation.

Analyzing Resources From an Endevor Remote System

When you work with resources that are hosted on an Endever Remote System, security restrictions might prevent you from seeing the contents of those resources in the Editor view. **IBM AD Analyze** is able to generate all the analyses for those resources, but to view their contents in the **Editor** view, you must set up a connection to the Endevor Remote System.

When the users work with mainframe resources of Endevor type, in order to view the contents of the resources in the Editor view they must set up a CARMA connection.

Note: To set up a CARMA connection, you need to have the CARMA feature installed either on RDz or IDz platform. During the installation process of **IBM AD Analyze Client** the **Application Discovery integration with Common Access Repository Manager (CARMA)** feature needs to be selected.

Let's assume you already have a project with sources brought from a remote mainframe system using the Endevor feature. You generate a graph, you select an element from the graph, and you want to see the contents of the selected resource in the Editor view. Right-click and from the shortcut menu select **View Analysis Source**.

If you didn't create a CARMA connection to the remote mainframe system, a warning message informs you that the selected resource cannot be accessed.

If you created one or several CARMA connections to the remote mainframe system, a list of available connections is displayed. Select which connection you want to use in order to view the contents the selected resource in the Editor view. The tooltip of the Editor view displays the details of the CARMA connection that was used to show the contents of the resource.

Below are the steps that must be taken in order to view the resource contents in the **Editor** view.

This task presents the steps that must be performed in order to see in the Editor view a resource from a remote Endevor system.

1. Create a project in **IBM AD Build** and bring resources from the remote Endevor system:
   a) In **IBM AD Build Client**: Create a project.
   b) In **IBM AD Build Configuration**:
      - Create a z/OS connection.
      - Associate the z/OS connection to your project.
      - Configure the z/OS connection for the Endevor system (the ENDEVOR Info tab).
   c) In **IBM AD Build Client**:
      - Bring resources from the remote Endevor system using the defined z/OS connection;
      - Build your project. For details on how to create a project in **IBM AD Build**, and how to bring resources from Endevor systems, see `IBM AD Build Client and Server User Guide`.

   Once you completed these steps, when you open your **IBM AD Analyze Client** the new project is available in the projects list in the **Explore projects** tab of AD Explore view.
2. In your **IBM AD Analyze Client**, define a CARMA connection to the remote Endevor system. When you set up the CARMA connection, make sure that to choose a user with access rights. For details on how to create a connection to a remote system, see the `IDz` or `RDz` documentation. For details on the supported versions, see `Prerequisite Software and Configurations`.
3. In your **IBM AD Analyze Client**, locate the wanted resource using the **Explore Project** tab and double-click it to display its contents in the Editor view. Alternatively you can generate a graph and select from the graph the element you want see in the **Editor** view. Right-click and from the pop-up menu select **View Analysis Source**.

Analyzing resources through APIs

Generating call chain analysis results by using API URLs

1. If the authentication and authorization feature is enabled with the Authentication Server (DEX) installed and configured, the **`passwordConnector`** parameter must be added in the `dex.yaml` file. Also, a security token is needed. For more information, see step 1.“4” on page 112 in topic “STEP 4. Configuring Authentication Server (DEX)” on page 111.
2. If OrientDB SSL is enabled, add the following properties in file *<IBM ADDI Installation Folder>*`/IBM AD Web Services/wlp/usr/servers/ad_server/jvm.options`:
   - The path of the OrientDB certificates.
   - The passwords of the certificates.

   Example

```
-Dclient.ssl.enabled=true
-Dclient.ssl.keyStore=C:\certificates\orientdb.ks
-Dclient.ssl.keyStorePass=password
-Dclient.ssl.trustStore=C:\certificates\orientdb.ks
-Dclient.ssl.trustStorePass=password
```

Note: You can enter an encrypted **Keystore Password**. For more information, see "Obtaining an encrypted keystore password" on page 209.

After the file is edited and saved, restart IBM Application Discovery WebSphere Liberty Profile Service.

The call chain web service runs on the Liberty server. After the call chain REST API is enabled, you can run requests to generate call chain analysis results by using URLs with other tools, for example, Postman. For more information about call chain analysis in IBM AD Analyze Client, see section "Miscellaneous Reports" on page 374.

Enter one of the following endpoints with the corresponding request body, and select the corresponding request method.

Endpoint	Request body	Request method
To conduct call chain analysis for one or more programs, you can use the following endpoint. Details can be specified in the request body. `libertyURL/callchain/` **Example** `http://localhost:9080/callchain/`	**Example** `{ "direction" : "forward", "projectName" : "Regression", "depth" : "3", "programsList" : [{ "prgName" : "DACI", "prgType" : "" }, { "prgName" : "ZZCU0013", "prgType" : "COBOL" }, { "prgName" : "DACD", "prgType" : "" }, { "prgName" : "ABDRAC01", "prgType" : "" }, { "prgName" : "COBMOD1", "prgType" : "" }, { "prgName" : "U11RHC", "prgType" : "ASSEMBLER" }, { "prgName" : "BONOUTOT", "prgType" : "" }] }`	POST
To conduct call chain analysis for one or more programs, you can use the following endpoint which includes the *showIncludes=true* query to display also the program-copybook relationship. Details can be specified in the request body. `libertyURL/callchain? showIncludes=true/` **Example** `http://localhost:9080/callchain? showIncludes=true/`	**Example** `{ "direction" : "forward", "projectName" : "Regression", "depth" : "3", "programsList" : [ { "prgName" : "A51DBDWN", "prgType" : "" }, { "prgName" : "D284PRI", "prgType" : "" } ] }`	POST

Endpoint	Request body	Request method
To conduct call chain analysis for one program per request, you can use the following endpoint. No request body is required. If the program type (*prgType*) is not specified for a program, analysis for all the programs with that name (*prgName*) will be returned. `libertyURL/callchain/withParameters?direction=callchainDirection&projName=projectName&depth=depthNumber&prgName=programName&prgType=programType` **Example** `http://localhost:9080/callchain/withParameters?direction=forward&projName=Regression&depth=3&prgName=ZZCU0013&prgType=COBOL`		GET
To check the programs that access the specified copybooks, you can use the following endpoint. `libertyURL/callchain/includes` **Example** `http://localhost:9080/callchain/includes`	`{ "$schema": "http://json-schema.org/draft-04/schema#", "description": "The programs input schema for the callchain includes/copybooks analysis API.", "type": "object", "properties": { "prjName": { "type": "string", "description": "The name of the project on which the analysis will be made." }, "includes": { "type": "array", "description": "An array which contains the includes names for the analysis.", "items": [{ "type": "object", "properties": { "inclName": { "type": "string", "description": "The name of a copybook/include" }, "inclType": { "type": "string", "description": "The source type of a copybook/include", "default": "" } }, "required": ["inclName"] }] } }, "required": [`	POST

Endpoint	Request body	Request method
	<pre>"prjName", "includes"] }</pre> **Example** <pre>{ "prjName": "Regression", "includes": [{ "inclName": "ACP1000R", "inclType": "COBOL" }, { "inclName": "#K5003ZA", "inclType": "" }] }</pre>	
To check the programs that access one specified copybook per request, you can use the following endpoint. No request body is required. If the copybook type (*inclType*) is not specified for a copybook, analysis for all the copybooks with that name (*inclName*) will be returned. <pre>*libertyURL*/callchain/includes/ withParameters? prjName=*projectName* &inclName=*copybookName* &inclType=*copybookType*</pre> **Example** <pre>http://localhost:9080/callchain/ includes/withParameters? prjName=Regression &inclName=ACP1000R &inclType=COBOL</pre>		GET

After the request is sent, call chain analysis results are generated and displayed. See the following example of analysis results.

Example

<pre>{
"longestChainLength": 3,
"noOfPrgsNeverCalled": 3,
"prgsDetails": [
{
"chains": [
"DACI<-NCP-ENTER-RP<-DAKL<-NCD-PF9-RP"
],
"noOfChains": 1,
"prgName": "DACI",
"type": "ADS Dialog"
},
{
"chains": [
"ZZCU0013:UTCBTRC<-ZZCU0013",
"ZZCU0013:UTCOTRC<-ZZCU0013",
"ZZCU0013:UTMSDEV<-ZZCU0013:UTCBTRC<-ZZCU0013",
"ZZCU0013:UTMSDEV<-ZZCU0013:UTCOTRC<-ZZCU0013",
"ZZCU0013:UTMSTRC<-ZZCU0013:UTCBTRC<-ZZCU0013",</pre>

```
                "ZZCU0013:UTMSTRC<-ZZCU0013:UTCOTRC<-ZZCU0013"
            ],
            "noOfChains": 6,
            "prgName": "ZZCU0013",
            "type": "COBOL"
        },
        {
            "chains": [
                "ABDRAC01<-ACRFUTST<-ZZCU0013:UTCBTRC<-ZZCU0013"
            ],
            "noOfChains": 1,
            "prgName": "ABDRAC01",
            "type": "COBOL"
        },
        {
            "chains": [
                "U11RHC:DTGDGBI<-U11RHC:GCBIAS",
                "U11RHC:SCAN<-U11RHC:DTGDGBI<-U11RHC:GCBIAS",
                "U11RHC:SCAN<-U11RHC:FNDREAN"
            ],
            "noOfChains": 3,
            "prgName": "U11RHC",
            "type": "ASSEMBLER"
        }
    ]
}
```

The call chain analysis results that are generated and displayed including program-copybook relationship are based on the following schema:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "The output schema for the callchain analysis API.",
    "type": "object",
    "properties": {
        "longestChainLength": {
            "type": "integer",
            "description": "The length of the longest chain."
        },
        "noOfPrgsNeverCalled": {
            "type": "integer",
            "description": "The number of programs that are never called or that are never
calling other program."
        },
        "prgsDetails": {
            "type": "array",
            "description": "The result list with details for each program returned by the
callchain analysis.",
            "items": [
                {
                    "type": "object",
                    "properties": {
                        "chains": {
                            "type": "array",
                            "description": "The list of chains returned for a program which
shows the usages or references that a program has.",
                            "items": [
                                {
                                    "type": "string",
                                    "description": "A chain of a program."
                                }
                            ]
                        },
                        "includes": {
                            "type": "array",
                            "description": "The list of accessed includes/copybooks returned
for a program",
                            "items": [
                                {
                                    "type": "string",
                                    "description": "An include/copybook accessed by a program"
                                }
                            ]
                        },
                        "noOfChains": {
                            "type": "integer",
                            "description": "The number of chains returned for a program"
                        },
                        "prgName": {
                            "type": "string",
```

```
                            "description": "The name of a program."
                        },
                        "type": {
                            "type": "string",
                            "description": "The type of a program."
                        }
                    },
                    "required": [
                        "chains",
                        "includes",
                        "noOfChains",
                        "prgName",
                        "type"
                    ]
                }
            ]
        }
    },
    "required": [
        "longestChainLength",
        "noOfPrgsNeverCalled",
        "prgsDetails"
    ]
}
```

Example result:

```
{
  "longestChainLength": 3,
    "noOfPrgsNeverCalled": 3,
    "prgsDetails": [
        {
            "chains": [
                "D284PRI:D284PRI->APKDAT"
            ],
            "includes": [
                "DTCOMM",
                "UBLOCK",
                "CDBZERO"
            ],
            "noOfChains": 1,
            "prgName": "D284PRI",
            "type": "ASSEMBLER"
        },
        {
            "chains": [
                "A51DBDWN->IOMOD",
                "A51DBDWN->ZZCANCEL",
                "A51DBDWN->ZZDATE65",
                "A51DBDWN->ZZDATEGT"
            ],
            "includes": [
                "A51COMPR",
                "ACP1000R"
            ],
            "noOfChains": 4,
            "prgName": "A51DBDWN",
            "type": "COBOL"
        }
     ]
}
```

The call chain copybook analysis results follow the following schema:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "The output schema for the Includes -> Programs analysis.",
    "type": "object",
    "properties": {
        "avgNoOfPrgsAccessingOneIncludd": {
            "type": "number",
            "description": "The average number of programs accessing one include/copybook."
        },
        "includesDetails": {
            "type": "array",
            "description": "The result list with details for each include returned by the
Includes -> Programs analysis.",
            "items": [
                {
```

```
                    "type": "object",
                    "properties": {
                        "allAccesses": {
                            "type": "integer",
                            "description": "The total number of accesses that the current
include/copybook has."
                        },
                        "distincPrgs": {
                            "type": "integer",
                            "description": "The total number of distinct programs accessing one
include/copybook."
                        },
                        "includeName": {
                            "type": "string",
                            "description": "The current include/copybook name."
                        },
                        "includePrgsDetails": {
                            "type": "array",
                            "description": "The list of programs details that accesses the
current include/copybook.",
                            "items": [
                                {
                                    "type": "object",
                                    "properties": {
                                        "occurrences": {
                                            "type": "integer",
                                            "description": "The number of accesses that a
program has for the current include/copybook."
                                        },
                                        "prgName": {
                                            "type": "string",
                                            "description": "The name of the program."
                                        },
                                        "prgType": {
                                            "type": "string",
                                            "description": "The source type of the program."
                                        }
                                    },
                                    "required": [
                                        "occurrences",
                                        "prgName",
                                        "prgType"
                                    ]
                                }
                            ]
                        },
                        "includeType": {
                            "type": "string",
                            "description": "The type of the current include/copybook."
                        },
                    },
                    "required": [
                        "allAccesses",
                        "distincPrgs",
                        "includeName",
                        "includePrgsDetails",
                        "includeType"
                    ]
                }
            ]
        },
        "maxNoOfPrgsAccessingOneIncllud": {
            "type": "integer",
            "description": "The maxmimum number of programs accessing on include/copybook."
        },
        "minNoOfPrgsAccessingOneIncllud": {
            "type": "integer",
            "description": "The minimum number of programs accessing on include/copybook."
        },
        "noOfPrgsAccessingIncludes": {
            "type": "integer",
            "description": "The total number of programs accessing includes/copybooks."
        }
    },
    "required": [
        "avgNoOfPrgsAccessingOneIncllud",
        "includesDetails",
        "maxNoOfPrgsAccessingOneIncllud",
        "minNoOfPrgsAccessingOneIncllud",
        "noOfPrgsAccessingIncludes"
```

```
    ]
}
```

Example result:

```
{
   "avgNoOfPrgsAccessingOneIncludd": 2.0,
    "includesDetails": [
        {
            "allAccesses": 2,
            "distincPrgs": 2,
            "includeName": "#K5003ZA",
            "includePrgsDetails": [
                {
                    "occurrences": 1,
                    "prgName": "LWOUT1:INIT3",
                    "prgType": "PL1"
                },
                {
                    "occurrences": 1,
                    "prgName": "LWOUT1:NIT1",
                    "prgType": "PL1"
                }
            ],
        "includeType": "PL1"
        },
        {
            "allAccesses": 2,
            "distincPrgs": 2,
            "includeName": "ACP1000R",
            "includePrgsDetails": [
                {
                    "occurrences": 1,
                    "prgName": "A50DBDWN",
                    "prgType": "COBOL"
                },
                {
                    "occurrences": 1,
                    "prgName": "A51DBDWN",
                    "prgType": "COBOL"
                }
            ],
        "includeType": "COBOL"
        }
    ],
    "maxNoOfPrgsAccessingOneIncludd": 2,
    "minNoOfPrgsAccessingOneIncludd": 2,
    "noOfPrgsAccessingIncludes": 4
}
```

Handling annotations by using API URLs

1. If the authentication and authorization feature is enabled with the Authentication Server (DEX) installed and configured, the **passwordConnector** parameter must be added in the `dex.yaml` file. Also, a security token is needed. For more information, see step 1.“4” on page 112 in topic “STEP 4. Configuring Authentication Server (DEX)” on page 111.
2. If OrientDB SSL is enabled, add the following properties in file *<IBM ADDI Installation Folder>*`/IBM AD Web Services/wlp/usr/servers/ad_server/jvm.options`:
   - The path of the OrientDB certificates.
   - The passwords of the certificates.

   Example

   ```
   -Dclient.ssl.enabled=true
   -Dclient.ssl.keyStore=C:\certificates\orientdb.ks
   -Dclient.ssl.keyStorePass=password
   -Dclient.ssl.trustStore=C:\certificates\orientdb.ks
   -Dclient.ssl.trustStorePass=password
   ```

   Note: You can enter an encrypted **Keystore Password**. For more information, see “Obtaining an encrypted keystore password” on page 209.

After the file is edited and saved, restart IBM Application Discovery WebSphere Liberty Profile Service.

The annotation web service runs on the Liberty server. After the annotation REST API is enabled, you can run requests to handle annotations by using URLs with other tools, for example, Postman. For more information about handling annotations in IBM AD Analyze Client, see section "Handling Annotations" on page 456.

Enter one of the following endpoints with the corresponding request body.

The create endpoint: *libertyURL*/annotations/project/create/*ProjectName*

When the annotation property of a project is null, this endpoint can be used to create annotations. Otherwise, use the update endpoint. You must specify the annotation keys and values that you want to create in the request body. See the following example:

URL:

```
http://localhost:9080/annotations/project/create/Regression
```

Body:

```
{
    "testare": "noua_testare",
    "testing": 43,
    "si mai multa testare": "sdasdasdasdasdasdasdasdasda",
    "bababa": "dsadasdas"
}
```

The update endpoint: *libertyURL*/annotations/project/update/*ProjectName*

This endpoint can be used to update existing annotations. If the annotation property of a project is null, this endpoint can also be used to create annotations. You must specify the annotation keys and values that you want to update in the request body. See the following example:

URL:

```
http://localhost:9080/annotations/project/update/Regression
```

Body:

```
{
    "1": "update_test",
    "2": "update_test_test",
    "3": "test_test_test",
    "4": 45
}
```

The delete endpoint: *libertyURL*/annotations/project/delete/*ProjectName*

This endpoint can be used to delete existing annotations. You must specify the annotation keys that you want to delete in the request body. The annotation values are optional. See the following example:

URL:

```
http://localhost:9080/annotations/project/delete/Regression
```

Body:

```
{
    "1": "",
    "2": "",
    "3": ""
}
```

The get endpoint: *libertyURL*/annotations/project/get/*ProjectName*

This endpoint returns the annotations that are set at the project level. See the following example:

URL:

```
http://localhost:9080/annotations/project/get/Regression
```

After a create, update, or delete request is sent, a message is displayed to confirm the result. See the following message examples:

```
Create annotations operation: successful
```

```
Update annotations operation: successful
```

```
Delete annotations operation: successful
```

The following example shows the response to a get request:

```
{
    "annotations_key_1": "annotations_value_1",
    "annotations_key_2": "annotations_value_2",
    "annotations_key_3": "annotations_value_3",
    "annotations_key_4": "annotations_value_4"
}
```

EWM to AD Integration

Introduction

Important: As of version 7.0, IBM Rational® Team Concert® (RTC) has been renamed to IBM Engineering Workflow Management (EWM). **RTC to AD Integration Tool** (RTC2AD) has not been renamed and it is used even if EWM is installed.

A growing need exists to provide to IBM Application Discovery for IBM Z (IBM AD) users a way to import source files from their EWM source repository into the appropriate AD project sub folders.

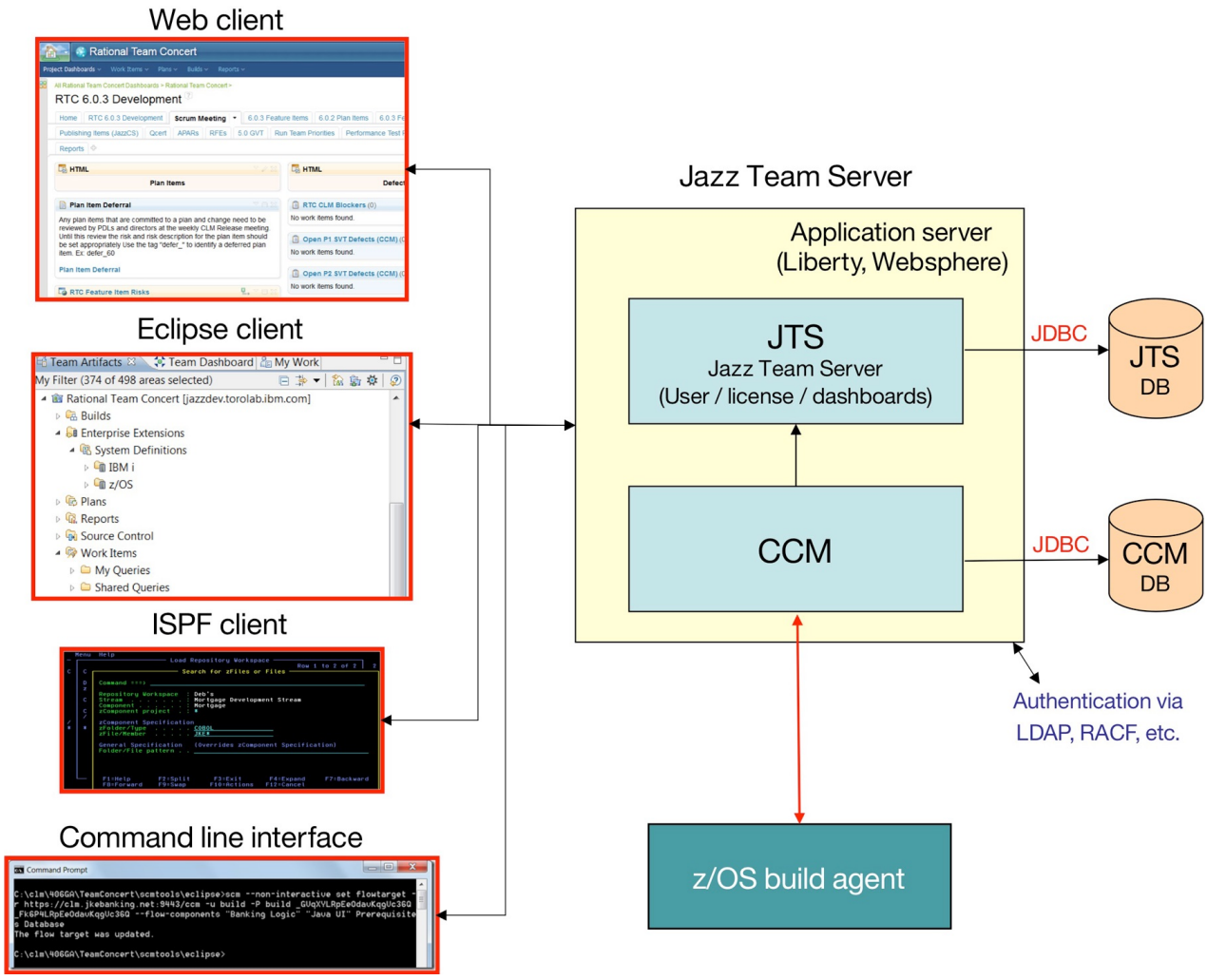
Below left, for example, is an EWM source repository that is on a remote Jazz® team server. Below right, is a set of sub folders from a sample AD project. The minimum requirement is to allow the AD user to connect to the EWM repository, select some or all of the source files, classify them based on source type, and finally copy them into the correct AD sub folder.



The **RTC to AD Integration Tool** (RTC2AD) provides a solution to this requirement. It gives an overview of EWM source management as it applies to z/OS, and how this tool functions to integrate it with AD source analysis.

EWM Overview

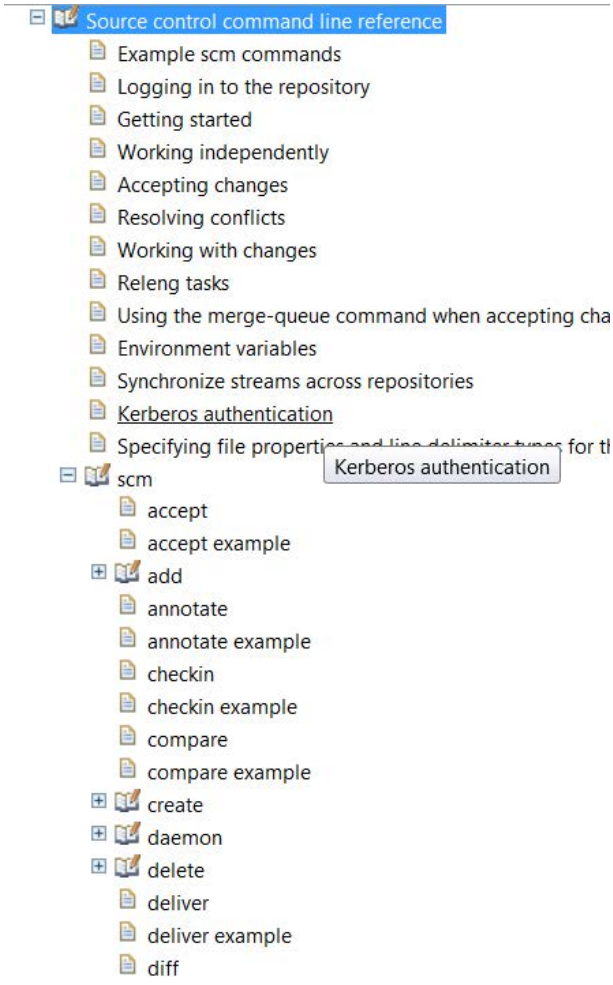
Note: All of the following diagrams, depicting facets of EWM, are copied from the developer works article.



Regarding the preceding diagram, the users need to use one of the EWM client options to work with their source repository stored on a Jazz Team server. The **RTC2AD** tool uses the **SCM command line interface** (CLI) of EWM, depicted at the lower left corner of the diagram.

On Windows™, the EWM CLI is named **scm.exe**. For a complete reference, including examples, see Source control command line reference.

It provides nearly every function that is available through the EWM Eclipse Client and so allows scripted interactions with EWM. This tool is the core of the script that provides **RTC2AD** integration.

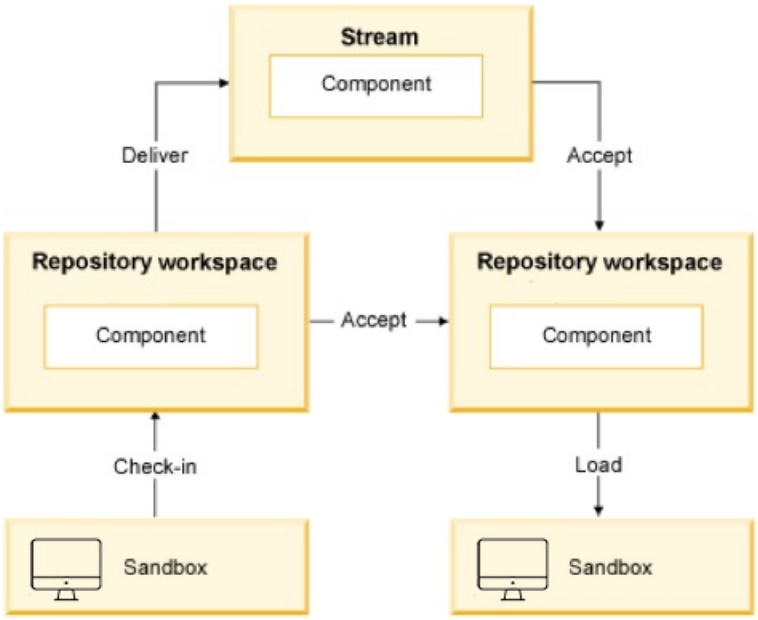


Source Management on EWM

EWM provides many different functions, but for AD integration purposes, it concentrates on its source management aspects, and more specifically on **z/OS source management**.

Several EWM objects that are part of this process:

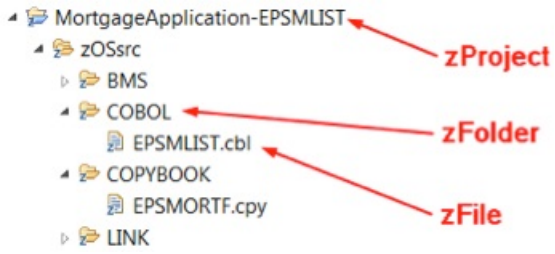
- **Stream** - represents a specific version, branch, or release of a collection of software components.
- **Component** - represents a collection of related software objects, such as a COBOL source library and COBOL copybook library. Together can be used to build a software component.
- **Repository Workspace** - represents a container object in EWM that can be loaded with components from a particular stream.
- **Sandbox** - represents a set of folders in the local file system where the user can load files from the workspace, make changes, and then check-in the changed files back to the repository workspace.



z/OS Specific EWM Objects

Special versions of EWM source control objects that apply specifically to z/OS:

- **zProject** - represents a type of EWM Eclipse project that contains a z/OS software project.
- **zComponent** - represents a specialized type of EWM component that contains zFolders.
- **zFolder** - is an EWM Eclipse folder that represents a z/OS PDS or PDS(E).
- **zFile** - is a file in a zFolder that represents a z/OS PDS(E) member. The zFile can be saved with a file extension to indicate its source type. In the example that follows, you can see a COBOL (`cbl`) file and a copybook (`cpy`) file. This presence of a file extension can save much time when the tool is attempting to classify the source type before it copies it to the correct AD project sub folder.



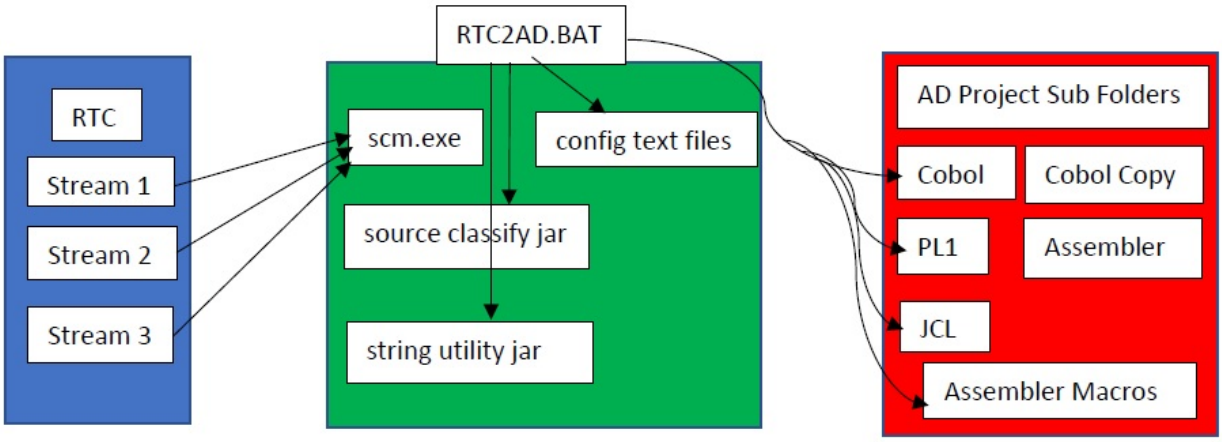
RTC to AD Integration Tool - RTC2AD

The RTC2AD integration tool is a Windows™ script, **rtc2ad.bat**, that makes calls to several functions provided by the EWM CLI program **scm.exe**. The IBM tool is distributed with the EWM Eclipse client (and possibly with other products). The assumption is that every EWM user has access to this tool.

The script also calls two Java™ JAR files, one to provide string parsing and tokenization, and the second to help with source type classification. Finally, a configuration file is used to customize the entire process.

- **com.ibm.dmh.scan.classify.jar** - scans a file on the local file system and determine the source type.
- **com.ibm.rtc2ad.jar** - performs utility functions like string manipulations that are too difficult or awkward to code in the Windows™ scripting language.
- **configuration text file** - is a text file that provides all parameters that are needed to connect to the EWM jazz server. It decides which EWM streams to retrieve, how to classify the files contained in them, and where to store them in the AD project sub folders.

Since a version of **scm.exe** is available for UNIX™ and Linux™, it is possible to adapt this tool to run on these platforms as well.

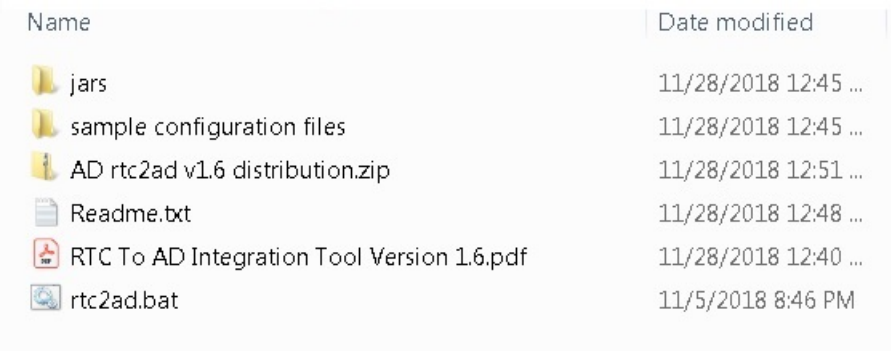


Installing and Configuring the RTC2AD Tool

Note: Make sure that you have the latest version of the RTC2AD tool. The latest version of the RTC2AD tool is compatible with all supported versions of IBM Application Discovery for IBM Z.

Follow these steps to install and configure the RTC2AD tool:

1. Extract the latest version of RTC2AD tool into any appropriate empty folder.



2. Make copies of the sample configuration files, rename them if you want and save them. It is convenient if they are all saved in the same folder.
3. Edit the main configuration file, originally named `config.txt` in the distribution. For more information, see "Configuration File Keyword Refence" on page 479 or refer to the comments in the main configuration file.
4. Edit all the other configuration files as necessary. In most cases, the sample files need some small adjustments. For more information, see "Configuration File Keyword Refence" on page 479 or refer to the comments in the main configuration file.
5. Copy both JAR files in the `jars` folder to an appropriate location. Ensure that **RTC2AD_CLASSPATH** in the main config file points to this folder.

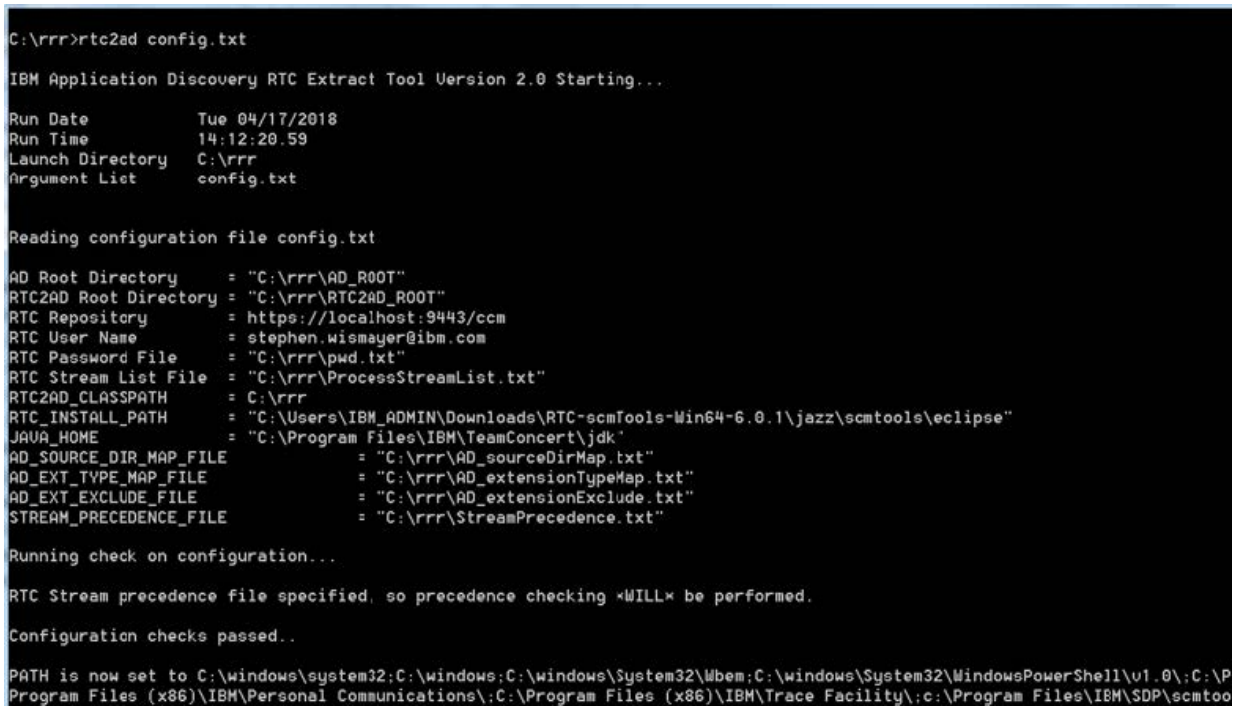
Running the RTC2AD Tool

Follow these steps to run the RTC2AD tool:

1. Open a command window and then use the cd command to proceed to the installation directory.
2. Considering that the configuration file name is `config.txt`, enter the following command: **rtc2ad config.txt**.

   Note: The tool starts and performs some initialization checks, then the main loop begins.

3. If the tool is unable to determine the type of a zFile, it is not copied to any AD project sub folder. Instead, the file name is written to the unresolved file, `unresolved.txt` in the directory in which the tool is running.



Full Load Versus Incremental Load

The RTC2AD tool operates in two distinct modes. The first mode is the default mode of operation. In this mode, it performs a Full Load of every source member that is contained in the EWM streams, which are specified by the configuration file. It is recommended to use this mode when the source members are imported from a stream for the first time.

The second mode is the incremental load. In this mode, only source members that are part of EWM change sets within a specified date interval are retrieved and classified. It greatly reduces the number of EWM components and files that need to be processed and in turn, the time that is required to complete the run. Incremental load is enabled when one or both of the following keywords are specified in the configuration file:

- **CREATED_AFTER_DATE**
- **CREATED_BEFORE_DATE**

The format of the key value fields that are associated with these keywords is **YYYY/MM/DD**. For more information, see “Configuration File Keyword Refence” on page 479.

The next step would be to run a Full Load when you start a new project, and then schedule incremental loads regularly to pick up any changes on an ongoing basis.

Configuration File Keyword Refence

The configuration file is a text file that contains a set of keyword - key value pairs. Several keywords reference other text files that are used to set certain parameters. The configuration file name is the only command line parameter that needs to be provided when you start `rtc2ad.bat`. A sample configuration file is provided in the `sample configuration files` folder of the rtc2ad installation folder, along with samples for all other files that are needed to run the tool.

Keywords	Description
AD_ROOT_DIR	Represents the path to the IBM AD project folder where the source is loaded. It must exist and it must already contain the sub folders. All the expected source types must exist in the necessary sub folders. This folder would typically be created by the AD user before the source is imported from EWM. The expected source types are designated in the **AD_SOURCE_DIR_MAP_FILE**.
RTC2AD_ROOT_DIR	Represents the root directory that serves as EWM sandbox. The sandbox is a set of directories on the local file system where EWM files and metadata are saved by the EWM CLI **scm.exe**. It must be separated from the **AD_ROOT_DIR**. If it does not exist, it needs to be created.
RTC_REPOSITORY	Represents the URL of the EWM repository where the zProject files are stored. It is the same URL that is used when a repository connection is created in the EWM Eclipse Client.
RTC_USERID	Represents the user ID under which the tool will log on to the EWM repository server. This user ID must have the appropriate access rights to the EWM project area from which the files are retrieved.
RTC_PASSWORD_FILE	Represents a text file that contains the password of the user that will log on to the EWM repository server. The password is in clear text, so this file needs to have appropriate access restrictions in place. This file is necessary. Refer to the sample file `pwd.txt` in the sample configuration files folder.
RTC_STREAM_LIST_FILE	Represents the list of EWM streams that the tool needs to process. By default, the tool processes automatically every EWM component within the specified streams that contains z/OS source. If you need to restrict processing to a subset of the components within a stream, refer to **RTC_COMP_LIST_FILE** description. This file is required. Refer to the sample file `ProcessStreamList.txt` in the sample configuration files folder.
RTC2AD_CLASSPATH	Represents the directory that contains the two JAR files that are needed by the tool, the files are `com.ibm.dmh.scan.classify.jar` and `com.ibm.rtc2ad.jar`. These JAR files are distributed in the `jars` folder.
RTC_INSTALL_PATH	Represents the directory that contains the EWM CLI **scm.exe**. It is made available as part of the EWM Eclipse installation. It is required by `rtc2ad.bat`.
Java™_HOME	Represents the directory that contains the Java™ SDK or JRE installation.
RTC_COMP_LIST_FILE	Represents an extra file that can be used to restrict which EWM components are processed within the streams that are specified in **RTC_STREAM_LIST_FILE**. Only EWM components that are listed on **RTC_COMP_LIST_FILE** are processed. If the keyword is omitted, default behavior is to process every component in every stream specified in **RTC_STREAM_LIST_FILE**. Refer to the sample file `ProcessComponentList.txt` in the sample configuration files folder.
AD_SOURCE_DIR_MAP_FILE	Represents a text file that specifies a mapping between the source type and the AD project subfolder name. It must contain all the AD subfolder names for different types of source that the tool encounters. This file is required. The sample file that is provided is usually sufficient for most cases. It can be updated as necessary. Refer to the sample file `AD_sourceDirMap.txt` in the sample configuration files folder.

Keywords	Description
AD_EXT_TYPE_MAP_FILE	Represents a text file that specifies a mapping between a zFile (PDS member) extension and a source type. For example, any zFile with `cbl` extension can be mapped to COBOL. If the file extensions are reliable indicators, the mapping allows the source classification step to be skipped entirely, saving time. If the file does not have an extension, or the extension is not recognized, the tool attempts to classify the source by calling the Java™ scanner utility. This file is required. Refer to the sample file `AD_extensionTypeMap.txt` in the sample configuration files folder.
AD_EXT_EXCLUDE_FILE	Represents a text file that contains a list of file extensions that needs to be ignored. When the tool encounters an EWM file that ends with one of these extensions, the file is ignored and it isn't copied to the AD project. This file is required. Refer to the sample file `AD_extensionExclude.txt` in the sample configuration files folder. This exclusion process can save time by filtering out unusual or proprietary file types, which do not need to be analyzed, or cannot be analyzed by AD.
CREATED_AFTER_DATE	Represents an extra keyword. The key value format is **YYYY/MM/DD**. If set, this keyword causes RTC2AD to run in incremental mode and only import files that are included in EWM change sets for the specified streams after the date specified. It can be used together with **CREATED_BEFORE_DATE** to define a date range. Only files that are part of an EWM change set delivered during this date range are processed. If both the **CREATED_AFTER_DATE** and the **CREATED_BEFORE_DATE** keywords are omitted, then RTC2AD runs in Full Load mode. It means that every source file in the specified streams is retrieved, classified, and copied to the AD folders.
CREATED_BEFORE_DATE	Represents an extra keyword. The key value format is **YYYY/MM/DD**. If set, this keyword causes RTC2AD to run in incremental mode and only import files that are included in EWM change sets for the specified streams before the date specified. It can be used together with **CREATED_AFTER_DATE** to define a date range. Only files that are part of an EWM change set delivered during this date range are processed. If both the **CREATED_AFTER_DATE** and the **CREATED_BEFORE_DATE** keywords are omitted, then RTC2AD runs in Full Load mode. It means that every source file in the specified streams is retrieved, classified, and copied to the AD folders.
COPY_SOURCE	Represents an extra test flag that has the default value **Y**. The **Y** value indicates that all source members that can be classified need to be copied to the appropriate AD source type folder. The **N** value indicates that the file needs to be classified and not copied. The option is useful only in testing the source classification routine.
RUN_ONLINE	Represents an extra test flag that has the default value **Y**. The **Y** value indicates that the tool needs to connect to the EWM server and download the specified source files into the local EWM sandbox. The **N** value indicates that no connection needs to be made to the EWM server. The processing needs to be performed by using only the files already stored in the local EWM sandbox. The option is useful only in testing, saving time by not repeating downloads from the EWM server.

Keywords	Description
DELTA_CHECK	Represents an extra test flag that has the default value **Y**. The **Y** value indicates that the tool needs to run a file compare between the source file that is processed and the identically named file that is classified and stored in the target AD source type folder (assuming it exists). If the files are identical, then no copy is performed. The **N** value indicates that the file compare step is skipped and the file is unconditionally copied to the appropriate AD source type folder.

Processing Summary

To provide a better understanding of the RTC2AD tool, check the following summary of the processing steps that the tool performs:

1. Read the configuration file and ensure that all necessary keywords are set.
2. Verify the existence of AD sub folders. If necessary, create an EWM_AD sandbox directory.
3. Set the **PATH**, **CLASSPATH**, and **JAVA_HOME** environment variables.
4. Test the call to **scm.exe** and run the test call to Java™ utility to confirm that **PATH** and **CLASSPATH** environment variables are set correctly.
5. Log on to EWM Jazz Team Server with the user credentials sourced from config file.
6. Read the stream list file.
7. Follow these steps for each stream present in the list file:
   a. Create a temporary EWM workspace and load it with the EWM stream.
   b. Retrieve the list of all components in the stream from EWM, save in temporary working file.
   c. Follow these steps for each component in the stream:
      i) If `pass list` file exists, check whether the component is on it, else skip component.
      ii) If no `pass list` file exists, process every component in the stream.
      iii) Retrieve the list of all objects contained in the EWM component.
      iv) Filter out only the zFile names and save them in a local file.
      v) Load all the zFiles from EWM into the local file system sandbox.
      vi) Follow these steps for each zFile:
         a) Check for the file extension. If the file extension exists, check if the mapping to source type exists.
         b) If the file extension is on `exclude` list, ignore it and proceed with next file.
         c) In case that no file extension is present, or extension is not mapped to source, call Java™ source classifier utility.
         d) If classifier can't resolve the source, write the file name in the global `unresolved` file.
         e) If the source type is resolved, update the file list with the member's type.
         f) Map the resolved source type to the correct AD source type sub folder.
         g) Based on type, update the file list with the member's target AD project sub folder.
      vii) After each file in the component is processed, build a file of copy commands.
      viii) A copy command is generated only if the file source type is resolved and type maps to AD folder.
      ix) Run the batch of copy commands, copying the file from local sandbox to the correct AD sub folder.
      x) Repeat the loop to process the next EWM component defined in the stream.
8. When all EWM components in stream are processed, loop to the next stream specified in config.

9. When all specified streams are processed, delete the temporary EWM workspace.
10. Log off from EWM Jazz Team Server.

Chapter 10. Tutorials

API Extensibility Tutorial

This tutorial shows you how the analysis of an API call is supported by the extensibility feature with sample files. You can also learn how to enable API call analysis for your own projects.

API Extensibility Sample Files

To help you get started with the API extensibility feature, the following sample files are provided in the AD installation package:

COBOL program files

The sample COBOL programs that contain API calls. You can find the program files in the `Cobol` folder.

`SQLCALL1.cbl`

The main COBOL program. The SQLCALL1 program calls the SQLGET program to perform calls based on program IDs 1, 2, 3, and 4.

`SQLGET.cbl`

The API program that performs a call to retrieve the program name of a specific program according to the program ID. The information that the SQLGET program needs to determine the called program is stored in a database table.

Java utility files

The sample Java utilities that are used to resolve the API call in sample programs SQLCALL1 and SQLGET. You can find the utility files in the `\Java User Exit\src\com\ibm\ez\resolver\utility` folder.

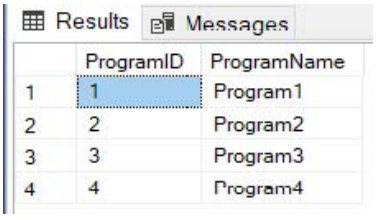
`ResolveCallUtility.java`

The Java utility that is used as a user exit to resolve the API call. This utility performs the following actions:

1. Parsing the input JSON file that is automatically generated at run time. For the schema of the input JSON file, see the `\Input and Output JSON Schema\utility-input-schema.json` file.
2. Using the SqlDataAccess utility to retrieve data.
3. Generating the output JSON file that contain the resolution of the API call. For the schema of the output JSON file, see the `\Input and Output JSON Schema\utility-output-schema.json` file.
4. Storing the input and output JSON files.

`SqlDataAccess.java`

The Java utility that is used to retrieve data from an SQL server according to the program ID. A database that is named DynamicCallPgms and a table that is named ProgramToCall are used. The following image shows the values in the ProgramToCall table:


Results Messages

	ProgramID	ProgramName
1	1	Program1
2	2	Program2
3	3	Program3
4	4	Program4

© Copyright IBM Corp. 2010, 2022

JAR files
The sample JAR files that are used to resolve the API call in sample programs SQLCALL1 and SQLGET: `sqljdbc42.jar` and `json-simple-1.1.1.jar`.

`Build.bat`
The sample file that can be used to compile sample Java utilities ResolveCallUtility and SqlDataAccess. You can find the file in the `Java User Exit` folder.

API extensibility configuration files
The sample JSON files that are used to configure the API extensibility settings. Two types of the configuration files are required to enable the API extensibility feature: API Config and User Exits Config. You can find the configuration files in the `AD Extensibility JSONs` folder.

`Api_Config.json`
The sample API Config configuration file that specifies the API program name, which is SQLGET, and the following two parameters that are required:

`programKey`
The value that is passed to the SQLGET program at run time through the *PGM-TOCALL-ID* variable to resolve the called program.

`data`
The value that is passed to the SQLGET program at run time through the *PGM-DATA* variable to resolve the called program.

`ue-config.json`
The User Exits Config configuration file that specifies the following user-exit-related settings:

- The ResolveCallUtility Java utility, which is used as a user exit to resolve the API call.
- The class path of the `sqljdbc42.jar` and `json-simple-1.1.1.jar` JAR files.
- The *###* notation that is replaced with a dynamically generated input JSON file name for each call at run time.
- The path to store the automatically generated input JSON files. The input files are parsed and stored by the ResolveCallUtility utility. This setting is optional and for debug purposes.
- The path to store the output JSON files that are generated by the ResolveCallUtility utility.

Input JSON files
The sample input files that are automatically generated at run time, and parsed by the sample Java utility ResolveCallUtility to retrieve the program IDs. You can find the input files in the `Input and Output JSON samples` folder.

Output JSON files
The sample output files that are generated by the sample Java utility ResolveCallUtility to specify the called program names. You can find the output files in the `Input and Output JSON samples` folder.

Schema JSON files
The files that describe the schemas of the sample input and output JSON files. You can find the schema files in the `Input and Output JSON Schema` folder.

Setting Up a Build with Sample Files

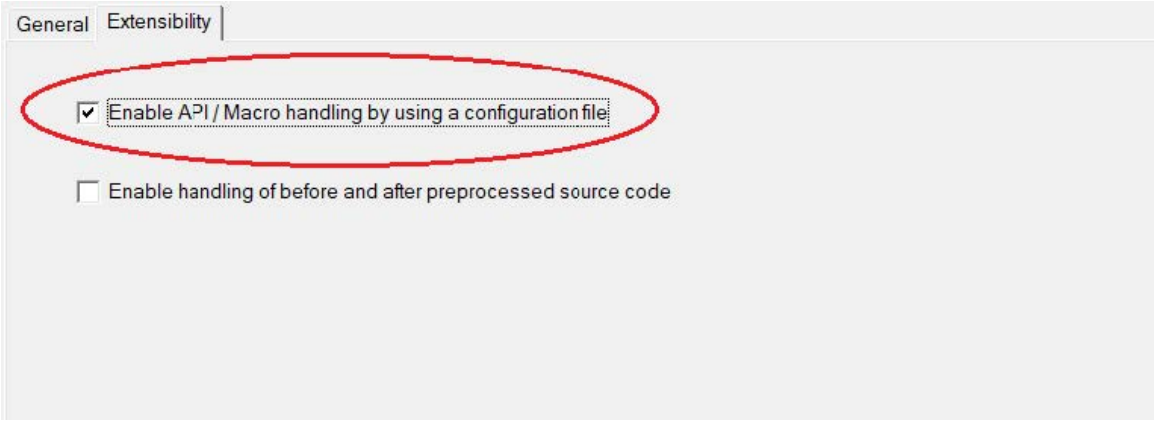
The API call analysis is supported by the AD extensibility feature. You can try out the feature by setting up a build with the API extensibility sample files. For more information about the sample files, see topic "API Extensibility Sample Files" on page 485.

Procedure

1. Create an AD project, and add the sample program files `SQLCALL1.cbl` and `SQLGET.cbl` into the project.
   - For more information about creating a project, see topic "Creating a project" on page 247.

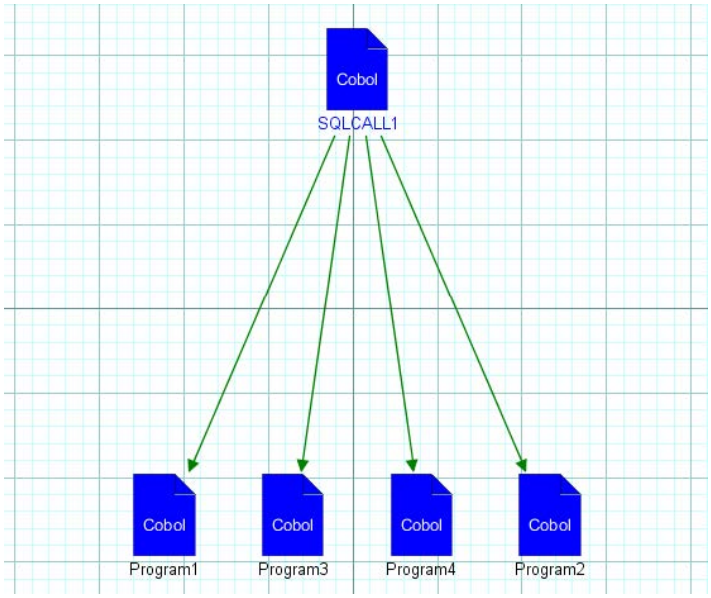
- For more information about adding files into a project, see topic “Adding Files to Project Folders” on page 257.

2. Enable the API extensibility feature. After the API extensibility feature is enabled, the `API Config` virtual folder is added into the project. This folder is used to store the extensibility configuration files.
   a. Click **Project > Settings > Extensibility**.
   b. In the **Extensibility** pane, select the "Enable API/Macro handling by using a configuration file" check box.

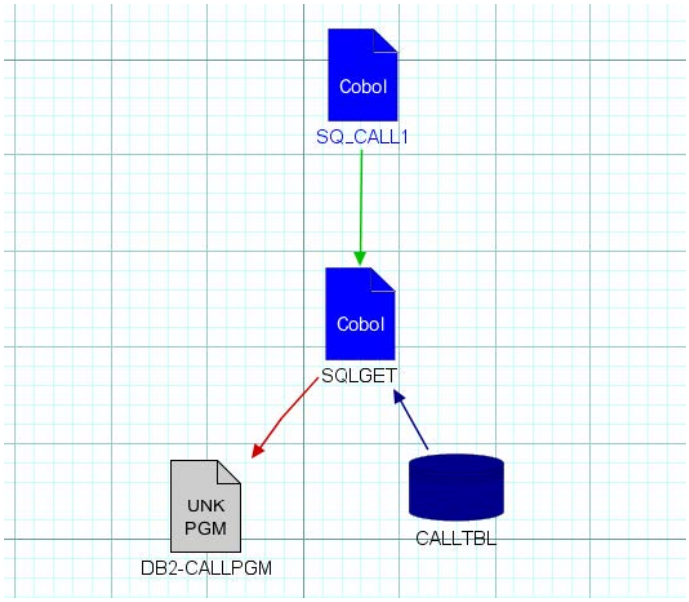


   c. Click **OK**.
3. Add the following two sample files into the `API Config` virtual folder of the project. The files are the API extensibility configuration files.
   - `Api_Config.json`
   - `ue-config.json`
4. Modify the paths in the `ue-config.json` file according to your settings.
   - Modify the class path of the sample JAR files according to your local environment.
   - Specify a path to store the input JSON files. This setting is optional and for debug purposes.
   - Specify a path to store the output JSON files.
5. Build the project. For more information about building an AD project, see topic “Building Projects” on page 264.

With the API extensibility feature enabled, after a project build is completed, the following callgraph is generated for the SQLCALL1 program. In addition to the standard output log file, an extensibility log file is generated by the build process and stored in the *<User folder name>*`\AD\comp\log` folder.



Without the API extensibility feature enabled, after a project build is completed, the following callgraph is generated for the SQLCALL1 program:



Extending from Sample Files to Your Projects

To support API call analysis for your own projects, you must create a user exit and API extensibility configuration files. The API extensibility sample files are provided as examples. Follow the steps in topic “Setting Up a Build with Sample Files” on page 486 to set up, but replace the sample files with your own project files. For more information about the sample files, see topic “API Extensibility Sample Files” on page 485.

Creating a user exit

The following functions must be implemented in your user exit:

- Reading and parsing an input JSON file.
- Resolving the API call by using the input JSON file and any additional metadata that is needed.
- Returning an output JSON file that specifies the resolution.

Creating API extensibility configuration files

Two types of the configuration files are required to enable the API extensibility feature: API Config and User Exits Config.

API Config
Specifies the API calls to be analyzed, the API call parameters, and for which one of these parameters, the values are needed.

User Exits Config
Contains a list of API calls and the path to your user exit.

Managing workspaces' access rights

Scenario 1: All Users Can See All Projects

Make sure you have **IBM AD** suite of applications installed and working;

Have at least a couple of projects created in **IBM AD Build Client**.

Publish the projects from **IBM AD Build Client** to **IBM AD Configuration Server**.

Workspaces can be Public, ACL (accessible by a specific list of domain users) or closed. In order to illustrate the way the workspace accessibility works let us have a look at the following scenario:

1. Create an environment. A Default workspace is created automatically and attached to the created environment. For this Default workspace set the accessibility to Public. Use only this workspace, do not create other workspaces.
2. Add a couple of projects to this Default workspace.

Access **IBM AD Analyze Client** and observe that all users can see all projects existing in the Default workspace.

Scenario 2: Users Cannot See Any Projects

Make sure you have **IBM AD** suite of applications installed and working;

Have at least a couple of projects created in **IBM AD Build Client**.

Publish the projects from **IBM AD Build Client** to **IBM AD Configuration Server**.

Workspaces can be Public, ACL (accessible by a specific list of domain users) or closed. In order to illustrate the way the workspace accessibility works let us have a look at the following scenario:

1. Create an environment. A Default workspace is created automatically and attached to the created environment. For this Default workspace set the accessibility to "Nobody". Use only this workspace, do not create other workspaces.
2. Add a couple of projects to the Default workspace.

Access **IBM AD Analyze Client**: the users cannot see any of the projects from the Default workspace.

Scenario 3: User with Access to a Workspace Can See All Projects from It

Make sure you have **IBM AD** suite of applications installed and working;

Have at least a couple of projects created in **IBM AD Build Client**.

Publish the projects from **IBM AD Build Client** to **IBM AD Configuration Server**.

Workspaces can be Public, ACL (accessible by a specific list of domain users) or closed. In order to illustrate the way the workspace accessibility works let us have a look at the following scenario:

1. Create an environment. A Default workspace is created automatically and attached to the created environment. For this Default workspace set the accessibility to ACL. Use only this workspace, do not create other workspaces.
2. Add a couple of projects to this Default workspace. Add a domain user to this Default workspace.

Access **IBM AD Analyze Client** with the user added in Step 2 and observe that the user can see the defined projects.

Scenario 4: User with Access to a Workspace Can See Only Projects from That Workspace

Make sure you have **IBM AD** suite of applications installed and working;

Have at least a couple of projects created in **IBM AD Build Client**.

Publish the projects from **IBM AD Build Client** to **IBM AD Configuration Server**.

Workspaces can be Public, ACL (accessible by a specific list of domain users) or closed. In order to illustrate the way the workspace accessibility works let us have a look at the following scenario:

1. Create an environment. A Default workspace is created automatically and attached to the created environment. For this Default workspace set the accessibility to "Nobody".
2. Create a second workspace, Workspace 2, set the accessibility for it as "ACL"(**A**ccess **C**ontrol **L**ist). Add one or two users to Workspace 2. Add a couple of projects to Workspace 2.

Access **IBM AD Analyze Client** with the user added in Step 2 and observe that the user can see only projects existing in Workspace 2 where he was granted access and cannot see/access the projects from the Default workspace.

Scenario 5: Users with Access to Specific Workspaces Can See Only Projects from their Assigned Workspace

Make sure you have **IBM AD** suite of applications installed and working;

Have at least a couple of projects created in **IBM AD Build Client**.

Publish the projects from **IBM AD Build Client** to **IBM AD Configuration Server**.

Workspaces can be Public, ACL (accessible by a specific list of domain users) or closed. In order to illustrate the way the workspace accessibility works let us have a look at the following scenario.

1. Create an environment. A Default workspace is created automatically and attached to the created environment. For this Default workspace set the accessibility to "Nobody".
2. Create a second workspace, Workspace 2, set the accessibility for it as "ACL"(**A**ccess **C**ontrol **L**ist). Add one or two users to Workspace 2. Add a couple of projects to Workspace 2.
3. Create a third workspace, Workspace 3, set the accessibility for it as "ACL". Add one or two users to Workspace 3 (different from the ones added to Workspace 2). Add a couple of projects to Workspace 3 (different from the ones added to Workspace 2).

Access **IBM AD Analyze Client** with any of the users added to Workspace 2: you can notice that the user can see only the projects existing in Workspace 2 where the user was granted access.

Access **IBM AD Analyze Client** with any of the users added to Workspace 3: you can notice that the user can see only the projects existing in Workspace 3 where he was granted access.

Chapter 11. Samples

Invoke an IBM utility program from JCL that calls Db2 related applications

The scenario of invoking an IBM utility program like **IKJEFT01** (or **IKJEFT1A**, **IKJEFT1B**) and passing the COBOL program that makes the actual Db2 calls in the SYSTSIN card is supported.

Example:

```
//jobname JOB USER=MY DB2ID
//GO EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=prefix.SDSNEXIT,DISP=SHR
//        DD DSN=prefix.SDSNLOAD,DISP=SHR
:
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
DSN SYSTEM (ssid)
RUN PROG (SAMPPGM) -
  PLAN (SAMPLAN) -
  LIB (SAMPPROJ.SAMPLIB) -
  PARMS ('/D01 D02 D03')
END
/*
```

There are other scenarios that may be supported only through extensibility API feature, if using another custom utility program that has a similar behavior as **IKJEFTxx**.

Rules File Example for Rule Based Analysis

Following is an example of how a file that contains rules for **Rule Based Analysis** must look like.

```
##for comments, use # at the beginning of the line. Lines with comments are ignored
##<number>: positive integer
##<identifier>: alphanumeric_text_with_0_spaces
##<path>: absolute path to a file on disk
##<class>: qualified class name for a rule
##<text>: any string of characters on one line. Line continuation (single backslash at the end
of line) can be used for multiple lines
##{key}: the value of the key as found in the file.
##[] in key names means optional


##includes are optional
#include.id = <number : larger than 0>
#include.key = <identifier>
#include.path = <path : an include file>

##input definition
##inputs are optional

#<input.id>.input.key = <identifier>
#<input.id>.input.name = <text : param name in gui>
#<input.id>.input.query = <path : to query file>
#<input.id>.input.firstColumnType = <numeric|string>

1.input.key = CobolPrograms
1.input.name = Cobol Programs
1.input.query = C:\\Rules\\CobolPrograms.txt
1.input.firstColumnType = numeric

##param definition
##parameters are optional
##if .query exists, .pattern is not used
##if .pattern not exists, default is unbound text

#<param.id>.param.key = <identifier>
```

© Copyright IBM Corp. 2010, 2022

```
#<param.id>.param.name = <text : param name in gui>
#<param.id>.param.default.value = <value>
#<param.id>.param.[pattern] = <pattern>
#<param.id>.param.[length] = <number>
#<param.id>.param.[value.lessThan] = <number>
#<param.id>.param.[value.greaterThan] = <number>
#<param.id>.param.[prefix] = <text : prefix for the chosen value>
#<param.id>.param.[suffix] = <text : suffix for the chosen value>

1.param.key = MaxINITIALIZESize
1.param.name = Max INITIALIZE record size
1.param.default.value = 3000

##group definition
##groups are optional

#<group.id>.group.name = <text : group name in gui>
#<group.id>.group.[parent] = [(0|{include.key}):]{group.id}
#<group.id>.group.[description] = <text : group description in gui>


##rule definition
##at least one of .class or .query must exist
##.query and .selectiveQuery can be used together for efficiency reasons.
##     Only one is executed depending on user selection

#<rule.id>.rule.name = <identifier: rule name>
#<rule.id>.rule.[sourceBased] = true|false (default true)
#<rule.id>.rule.[weight] = <number> (default 1)
#<rule.id>.rule.[query] = <path : to query file> (default none)
#<rule.id>.rule.[selectiveQuery] = <path : to selective query file> (default none)
#<rule.id>.rule.[class] = <class> (default none)
#<rule.id>.rule.[groups] = [(0|{include.key}):]{group.id}[, [(0|{include.key}):]{group.id}]*
(default none)
#<rule.id>.rule.[description] = <text: rule description> (default none)
#<rule.id>.rule.[params] = [(0|{include.key}):]{param.key}[, [(0|{include.key}):]{param.key}]*
(default none)
#<rule.id>.rule.[inputs] = [(0|{include.key}):]{param.key}[, [(0|{include.key}):]{param.key}]*
(default none)

# Groups

10.group.name=Performance Rules
10.group.description=These rules look up potential performance issues in the code

100.rule.groups=10
100.rule.name=INITIALIZE of large records
100.rule.description=INITIALIZE of very large records consumes a lot of CPU, consider the
alternative of using a direct MOVE
100.rule.sourceBased=true
100.rule.query=C:\\Rules\\InitializeRule.txt
100.rule.weight=1
100.rule.inputs=CobolPrograms
100.rule.params=MaxINITIALIZESize
```

Following are two examples of how the files that contain SQL queries must look like.

The query in the following example brings data for the rule "INITIALIZE of large records" from previous the example.

SQL Query Example:

```
SELECT DISTINCT TOP (100) PERCENT dbo.Programs.ProgramID, dbo.Programs.ProgramName,
dbo.Paths.PathStr,
                          dbo.OccurrencesStmt.StartRow, ISNULL
                         ((SELECT     0 AS Expr1
                             WHERE     (dbo.OccurrencesStmt.SourceType < 4)), - 1) AS IsCopy
FROM         dbo.StatementReference INNER JOIN
                      dbo.OccurrencesStmt ON dbo.StatementReference.OccurID =
dbo.OccurrencesStmt.OccurID INNER JOIN
                      dbo.Programs ON dbo.OccurrencesStmt.ProgID = dbo.Programs.ProgramID INNER
JOIN
                      dbo.Variables ON dbo.StatementReference.ResourceID = dbo.Variables.VarID
INNER JOIN
                      dbo.Paths ON dbo.OccurrencesStmt.PathID = dbo.Paths.PathID
WHERE     (dbo.StatementReference.StatementType = 24) AND (dbo.StatementReference.ResourceType
= 4) AND
                      (dbo.Variables.VarSize > {MaxINITIALIZESize}) AND
```

```
                        (dbo.Programs.ProgramID in ({CobolPrograms:-1}))
ORDER BY dbo.Programs.ProgramName, dbo.Paths.PathStr, dbo.OccurrencesStmt.StartRow
```

Following is an example of an input query for AllPrograms, as it is defined in the .properties file.

Input query for AllPrograms Example.

```
SELECT     TOP (100) PERCENT ProgramID, ProgramName
FROM         dbo.Programs
WHERE     (OccurID > 0) AND (ProgramTypeID = 1)
ORDER BY ProgramName
```

Following is an example of the sample files to be used to test the Rule Based Analysis in the Analyze Client.

1. Create a text file by copying and pasting the content of the above first example, and name it `RuleBasedAnalsis.properties`.
2. Create a text file by copying and pasting the content of the above second example, and name it `InitializeRule.txt`.
3. Create a text file by copying and pasting the content of the above third example, and name it `CobolPrograms.txt`.
4. Create a folder `C:\Rules`, and put the above three files to this folder.
5. Open the Eclipse, and click **Window** > **Preferences** > **Application Discovery** > **Local settings** > **Rules Based Analysis**. Check the **Overwrite global configuration**, and enter for the Rules: `C:\Rules\RuleBasedAnalsis.properties`. Then, select your **File encoding** to be UTF-8 or else, press **Apply**, and close the page.
6. In the Analyze Client, select a project, and run **Mainframe Reports** > **Miscellaneous Reports** > **Rule Based Analysis**. The wizard will be opened, and one rule "INITIALIZE of large records." can be displayed. Select the rule, and you can change the parameter values. Then, move it to the **Selected rules** box, press **Next**, and select some or all of the COBOL programs. Finally, press **Finish**, and the report will be created.

Log Configuration Files Example

`LogConfigurator.properties` allows the user to customize the amount of details that is generated about errors. `LogConfigurator.properties` is created by the user, is fully configurable, and must be placed in Eclipse's workspace folder as in: `<Local Workspace>\.metadata\.ez\`.

Following is an example of a `LogConfigurator.properties` file.

```
com.ez=info
com.ez.graphs.viewer.odb=debug
```

Following is an example of a `Log4j.properties` file.

```
log4j.rootLogger=info, stdout, R

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%-5p %d [%t] %c - %m\n

log4j.appender.R=com.ez.log4j.appender.EZRollingFileAppender

log4j.appender.R.FileName=/log/ez.log

log4j.appender.R.MaxFileSize=100MB
log4j.appender.R.MaxBackupIndex=10

log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%-5p %d [%t] %c - %m\n

log4j.logger.com.googlecode.protobuf.pro=info
log4j.logger.EZSourceConnectionService=error
log4j.logger.Unknown=error
log4j.logger.org.apache.zookeeper=error
```

```
log4j.logger.com.ez=info
```

Synchronize Members Configuration File Examples

The configuration file is intended to instruct **IBM AD Build Client** on whether it needs to update against specific libraries, where to add/remove the related members in/from the project (that is, which virtual folder to use) and also which type of members **IBM AD Build Client** must use when you add members. The basic assumption is that the specified libraries do not contain members that do not have to be added even though they are there.

Note: IBM AD works with a single version of a file in a given project, as it represents the state of a given application in one environment at a time.

The configuration file must contain the following parameters:

<Project name>, <Library type >, <Library name>, <Mapped virtual folder>, <Members type >, <z/OS> ,Application

Project name
The name of the project that needs to be synchronized with the mainframe system.

Library type
The type of the library from which members are added.

Library name
The name of the library against which the synchronization takes place.

Mapped virtual folder
The name of the project folder where the members are added/updated/deleted.

Members type
The type of the members that are synchronized. The allowed types are presented in the **Member types names** table.

z/OS
The name of the connection to the mainframe system, as defined in the **zOS** tab of the **IBM Application Discovery Build Configuration** window. For more information, see "The zOS Tab" on page 236.

Application
The name of the application as defined for ChangeMan ZMF in **zOS Configuration** screen.

Following are some examples for configuration files, based on Library types.

Note: Each line in the configuration file must contain a unique <Project name>, <Library type >, <Library name>, <Mapped virtual folder>, <z/OS> tuple. If not, the configuration file is invalidated. Also, if the synchronization process runs for more than one Library/Project, a unique line must be added for each project that needs synchronization.

To comment a line in the configuration file, add ***** at the beginning of that line.

PDS Library- <PROJECT>, PDS(MVS), ITLS.COBOL.II, zOS Cobol, COBOL_MVS, zOSCONN

Endevor- <PROJECT>, Endevor, SMPLTEST.EZLPROJ.EZLCOMP.COBOL.T, zOS Cobol, COBOL_MVS, zOSCONN

ChangeMan Baseline- <PROJECT>, SERENA CHANGEMAN BASELINE, EZL.SERENA.SYNC.EZLX.COB, zOS Cobol, COBOL_MVS, zOSCONN, EZLX

ChangeMan Package- <PROJECT>, SERENA CHANGEMAN PACKAGE, EZLX000020, zOS Cobol, COBOL_MVS, zOSCONN

Available library and SCM types- PDS(MVS), Endevor, SERENA CHANGEMAN BASELINE, SERENA CHANGEMAN PACKAGE

The synchronize process against Local/Remote Folders is supported. Each line in the configuration file must contain:

```
Project name in IBM AD Build Client, LOCAL_REMOTE, Full path of the folder where the sources
exist, Virtual folder in IBM AD Build Client, Members type within Virtual Folder, (optionally)
Filter the files found in the source folder (search criterias are separated by | )
```

Examples:

1. When not using a filter.

   Note: The entire folder's content is synchronized.

   ```
   MyProject, LOCAL_REMOTE, C:\IBM AD\Mainframe Sources\Local Sources, zOS Cobol, COBOL_MVS
   ```

2. When using a filter.

   Note: Only the files that are matched by the filter are synchronized.

   Examples:

   ```
   MyProject, LOCAL_REMOTE, C:\IBM AD\Mainframe Sources\Local Sources, zOS Cobol, COBOL_MVS,
   filter(*.cbl|*.cob)
   ```

   Where:

   - `*.cbl` - only the COBOL files that have the `*.cbl` extension are synchronized.
   - `*.cob` - only the COBOL files that have the `*.cob` extension are synchronized.

   ```
   MyProject2, LOCAL_REMOTE, C:\IBM AD\Mainframe Sources\Local Sources, zOS Cobol, COBOL_MVS,
   filter(a*|*b)
   ```

   Where:

   - a* - only the files that starts with letter **a** are synchronized.
   - *b - only the files that ends with letter **b** are synchronized.

3.

Member types names:

Virtual Folder	FileTypeName
AAuto Scheduling	AAUTO_SCHEDULING
AAuto Scheduling	AAUTO_DS_FLAG_REPORT
ADS Dialog	ADS_DIALOG
ADS Map	ADS_MAP
ADS Process	ADS
Assembler	ASSEMBLER
Assembler Include	ASM_INCLUDE
Assembler Macro	ASM_MACRO
BMS	BMS
Cobol IDMS	COBOL_IDMS
Cobol IDMS Record	COBOL_IDMS_RECORD
Cobol Include	COPY
Configuration	CSD
Configuration	IMST_PGM
Configuration	JCL_PGM

Virtual Folder	FileTypeName
Configuration	PGM_ALIAS
Data Area	LDA
DBD	DBD
IMS Map	MFS
JCL	JCL
JCL Control files	JCL_CTRL
JCL Include	JCL_INCLUDE
JCL Processes	JCL_PROCLIB
Log	LOG
MQ	MQ_CONFIG
Natural	NATURAL
Natural Include	NATURALINCLUDE
Natural Map	NATURALMAP
PL1	PL1
PL1 IDMS Record	PL1_IDMS_RECORD
PL1 Include	PL1_INCLUDE
PSB	PCB
Schema	SCHEMA
Subschema	SUBSCHEMA
zOS Cobol	COBOL_MVS
PreProc Before	EXTENS_PREPROC_BEFORE
PreProc MetaData	EXTENS_PREPROC_META
PreProc Config	EXTENS_PREPROC_CONFIG

Validation configuration files examples.

`ProjectMapping.txt`

This file is the configuration file for defining the Project Name, Serena Application Name, and the Subsystem that is used for the validation process. Valid means that the format is correct and the Project does exist.

```
<Project Name>, <Serena Application Name>, <Subsystem> (comma separated values)
```

`IncludesOrder.txt`

This file is the configuration file for defining the Baseline Libraries types and the order, of Cobol Includes locations. This configuration file is used later on while you set up the path for the Cobol Include folders.

```
<type>,<type1>,….<typen>
```

Note: It is EXTREMELY important to add the types in the order in which the include files must be looked after.

`FoldersMapping.txt`

This file is the configuration file for defining a mapping between a Logical Folder of the project and the type of a member that is part of the validation process. This configuration file is used during the validation stage of the Synchronize Members process.

```
<Member Type>,<Logical folder>
```

`ServicePort.txt`

This file is the configuration file for defining the Service's port. If you use a firewall, make sure that the port is added as an exception.

```
<Port Number>
```

Extensibility JSON/Configuration File Examples

Schema JSON files and examples of creating configuration JSON files are provided in this appendix.

Preprocessing Extensibility Examples

Examples are provided to show how metadata JSON files and configuration files must be created to integrate their business-specific cases in the analysis.

The following use case is for the situation when a user uses the `%Call using CICS XCTL` and #Copybook macros in a before file.

Before file example: PREPROC.bef

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID.
PREPROC.

       DATA DIVISION.
       WORKING-STORAGE SECTION.
        01  FIRST-NAME PIC X(09).
        01  COMMAREA-FOR-PROG2 PIC X(09).
       PROCEDURE DIVISION.
          MOVE 'TEST-NAME' TO FIRST-NAME.
      CALL 'PROG1'
                USING FIRST-NAME.
         %Call using CICS XCTL
             #Copybook
```

After file example: PREPROC.cbl

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID.
PREPROC.

       DATA DIVISION.
       WORKING-STORAGE SECTION.
        01  FIRST-NAME PIC X(09).
        01  COMMAREA-FOR-PROG2 PIC X(09).
       PROCEDURE DIVISION.
          MOVE 'TEST-NAME' TO FIRST-NAME.
          CALL 'PROG1'
                USING FIRST-NAME.
           EXEC CICS XCTL PROGRAM('PROG2')
               COMMAREA(COMMAREA-FOR-PROG2)
               LENGTH(38) END-EXEC.
          CALL 'PROG3'
                USING FIRST-NAME.
```

The following lines in the after file are from Copybook:

```
          CALL 'PROG3'
                USING FIRST-NAME.
```

JSON file used for mapping: PREPROC.meta

```
{
    "info":
    {
        "version":"1.0.0"
    },

    "metadata":
    [
        {
            "pathType": "PC",
            "beforePath":"C\\BeforeFolder\\PREPROC.bef",
            "afterPath":" C\\AfterFolder \\PREPROC.cbl",

          "diffResolution":
          [

            {
              "afterPos":
              {
                "startLine": 1,

                "endLine": 10

              },

              "beforePos": {
                "startLine": 1,

                "endLine": 10

              }

            },

            {
              "afterPos": {
                "startLine": 11,

                "endLine": 13

              },

              "beforePos":
              {
                "startLine": 11,

                "endLine": 11
              }

            },
                       {
              "afterPos":
              {
                "startLine": 14,

                "endLine": 15

              },

              "beforePos": {
                "startLine": 1,

                "endLine": 2
                },
              "type": "INCLUDE",
              "path": "C\\CopybooksFolder\\Copybook",
              "includeStmtPos": {

                "includeStmtPath": " C\\BeforeFolder\\PREPROC.bef",
                "includeStmtLine": 12

              }

            }

          ]
         }
```

```
      ]
}
```

Configuration file example

The configuration file contains the information of the mappings between before files, metadata files, after files, and the extensions for each type of the files:

```
"C\BeforeFolder1\" | "C\MetaFolder1\" | " C\AfterFolder1\" | ".bef" | ".meta" | ".cbl"
"C\BeforeFolder2\" | "C\MetaFolder2\" | "C\AfterFolder2\" | ".cbl" | ".meta" | ".after"
"C\BeforeFolder3\" | "C\MetaFolder3\" | "C\AfterFolder3\" | ".cbl" | ".meta" | ".after"
```

Extensibility preprocessing JSON schema

```
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "title": "AD Extensibility preprocessing definition file",
  "type": "object",

  "properties": {
    "info": {
      "type": "object",
      "properties": {
        "version": {
          "type": "string",
          "description": "Version of the preprocessing metadata format",
          "pattern": "^[0-9]+\\.[0-9]+\\.[0-9]+\\."
        }
      },
      "required": [ "version" ],
      "additionalProperties": false
    },

    "metadata": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "pathType": {
            "type": "string",
            "description": "Format of the before and after file paths: local or mainframe",
            "enum": [
              "PC",
              "MF"
            ]
          },
          "beforePath": {
            "type": "string",
            "description": "Path of the original source",
            "minLength": 1
          },
          "afterPath": {
            "type": "string",
            "description": "Path of the preprocessed source",
            "minLength": 1
          },
          "diffResolution": {
            "type": "array",
            "description": "Ordered list of correspondences between lines in the after file and lines in the before file",
            "items":
            {
              "oneOf": [
                        {
                            "$ref": "#/definitions/diffResolutionType1"
                        },
                        {
                            "$ref": "#/definitions/diffResolutionType2"
                        }
                    ]
            }
          }
        },

        "required": [ "pathType", "beforePath", "afterPath", "diffResolution" ],
        "additionalProperties": false
      }
    }
  },

  "required": [
    "info",
    "metadata"
  ],

    "definitions":
  {
    "position": {
      "type": "object",
      "description": "Interval of lines in the source file",
      "properties": {
        "startLine": {
          "type": "integer",
          "description": "Starting line of the interval",
          "minimum": 1
        },
        "endLine": {
          "type": "integer",
          "description": "Ending line of the interval",
```

```
          "minimum": 1

        }
      },
      "required": [ "startLine", "endLine" ],
      "additionalProperties": false
    },

    "diffResolutionType1":
    {
      "type": "object",
      "properties": {
        "type": {
          "type": "string",
          "description": "The type of line mapping, if a special mapping is required",
          "enum": [
            "INCLUDE"
          ]
        },
        "includeStmtPos" : {
          "description": "Position of the include statement",
          "type": "object",
          "properties":
          {
             "includeStmtPath": {
               "type": "string",
               "description": "Path of the statement occurence",
               "minimum": 1
             },
             "includeStmtLine": {
               "type": "integer",
               "description": "Line of the statement occurence",
               "minimum": 1

             }
          },
          "required": [ "includeStmtPath", "includeStmtLine" ],
          "additionalProperties": false
        },
        "path": {
          "type": "string",
          "description": "Path to the source of the lines, if a special type of mapping was specified",
          "minLength": 1
        },

        "beforePos": {
          "description": "Position in the before file source",
          "$ref": "#/definitions/position"
        },
        "afterPos": {
          "description": "Position in the after file source",
          "$ref": "#/definitions/position"
        }
      },

      "required": [ "type", "path", "beforePos", "afterPos" , "includeStmtPos"],
      "additionalProperties": false
    },

    "diffResolutionType2": {
      "type": "object",
      "properties": {
        "beforePos": {
          "description": "Position in the before file source",
          "$ref": "#/definitions/position"
        },
        "afterPos": {
          "description": "Position in the after file source",
          "$ref": "#/definitions/position"
        }
      },

      "required": [ "beforePos", "afterPos" ],
      "additionalProperties": false
    }
  }
}
```

API/Macro Call Extensibility Examples

API Call/Macro Extensibility Use Cases

The purpose of this chapter is to help users understand how the Configuration files must be created in order to integrate their business specific cases in the analysis.

API Programs for calling utilities Use case

This use case is created to cover the situation when the API programs are used to call a program that is defined through one of the initial call parameters. The following represents the **Sample1** Cobol program:

```
IDENTIFICATION DIVISION.
       PROGRAM-ID. Sample1.
      *
       ENVIRONMENT DIVISION.
      *
       DATA DIVISION.
      *
```

```
       WORKING-STORAGE SECTION.
        01 VAR1 PIC X(06).
        01 VAR2 PIC X(06).
        01 VAR3 PIC X(08).
        01 VAR4 PIC X(08).
        01 X PIC X(08).

       PROCEDURE DIVISION.
          MOVE  'VALUE1'  TO VAR1
        MOVE  'VALUE2'  TO VAR2
       CALL 'API1' USING VAR1, VAR2, VAR3, VAR4.
```

The **Sample1** Cobol program contains a call to a specific API program named **API1**.

In case a call to the **API1** program has to be interpreted as a call to the value of the first call parameter (**VALUE1**) , the following Configuration Files containing the rules for the new resolution are to be used.

Note: The `API_Config.json` file specifies that the fist parameter has to be saved with its value while for the rest of parameters there is no need for their values.

```
{
    "info":
   {
       "version":"5.0.4.1"
   },
    "extensions":
    [
        {
           "apiKey": "a1",
           "name": "API1",
           "type": "call",
           "parameters":
           [
              {
                 "position":1,
                 "label": "program",
                 "resolve": true
              },
              {
                 "position":2,
                 "label": "accessType",
                 "resolve": false
              },
           {
                "position":3,
                "label": "outputVariable",
                "resolve": false
              },
              {
                "position":4,
                "label": "inputVariable",
                "resolve": false
              }
           ]
        }
    ]
}
```

The `ue-config.json` file specifies the type of user exit file (for v5.0.4, only `.Json` file is supported) and the path where the user exit file is located.

```
{
  "schemaVersion":"1.0.1",
  "documentVersion":"1.0.1",
  "ueConfig" :[
   {"name": "uejson",
    "type":"file-json",
    "location":"location/of/the/file/which/contains/the/resolution.json",
    "appliesTo":["a1"]
   }
]
}
```

The `resolution.json` file specifies that the API1 call is replaced with a call to a Cobol program, it's name is resolved in the first parameter and parameters 2, 3 and 4 are only referred.

```
{
    "schemaVersion": "1.0.1",
    "documentVersion": "1.0.1",
    "entries": [{
        "input": {
            "apiKey": "a1",
            "apiText": "API1",
            "params": [{
                    "paramKey": "program",
                    "paramMode": "byValue"
                },
                {
                    "paramKey": "accessType",
                    "paramMode": "byRef"
                },
                {
                    "paramKey": "outputVariable",
                    "paramMode": "byRef"
                },
                {
                    "paramKey": "inputVariable",
                    "paramMode": "byRef"
                }
            ]
        },
        "resolutions": [{
            "action": "call",
                    "target": {
                "resourceType": "CobolProgram",
                "paramKey": "program",
                "paramMode": "byValue",
                "read": true
            },
            "resources": [{
                    "resourceType": "Variable",
                    "paramKey": "accessType",
                    "paramMode": "byRef",
                    "read": true
                },
                {
                    "resourceType": "Variable",
                    "paramKey": "outputVariable",
                    "paramMode": "byRef",
                    "read": true
                },
                {
                    "resourceType": "Variable",
                    "paramKey": "inputVariable",
                    "paramMode": "byRef",
                    "read": true
                }
            ]
        }]
    }]
}
```

API Programs for Data Access Use Case

This use case is created to cover the situation when the API programs are used to perform SQL Select and SQL Delete on specified fields from TABLE1 and TABLE2 SQL tables defined in the Cobol program. The following represents the **Sample2** Cobol program.

```
IDENTIFICATION DIVISION.
       PROGRAM-ID. Sample2.
      *
       ENVIRONMENT DIVISION.
      *
       DATA DIVISION.
       WORKING-STORAGE SECTION.

        EXEC SQL DECLARE TABLE1 TABLE
           ( COL11                          CHAR(7),
             COL12                         CHAR(10)
           )END-EXEC.
```

```
        EXEC SQL DECLARE TABLE2 TABLE
           ( COL21                          CHAR(7),
             COL22                         CHAR(10)
            )END-EXEC.

       01  VALUE1-DATA.
         03 VALUE1-DATA  PIC S9(6).
       01  VALUE2-DATA.
      03 VALUE2-DATA  PIC S9(6).
       01  FIRST-PARAM.
         03 ACCESS-ID PIC S9(6).

       PROCEDURE DIVISION.

        MOVE 'VALUE1' TO ACCESS-ID
        MOVE 'AAAAAA' TO VALUE1-DATA
        CALL 'API2' USING FIRST-PARAM
                          VALUE1-DATA.

        MOVE 'VALUE2' TO ACCESS-ID
        MOVE 'AAAAAA' TO VALUE2-DATA
        CALL 'API2' USING FIRST-PARAM
                          VALUE2-DATA.
```

The first call to the API2 program has to be interpreted as an SQL Select performed on fields *COL11* and *COL12* from **TABLE1 SQL** table. This action is specified by the value of the second API call parameter **VALUE1**.

The second call to the API2 program has to be interpreted as an SQL Delete performed on fields from **TABLE SQL** table. This action is specified by the value of the second API call parameter **VALUE2**.

The second call to the API2 program has to be interpreted as an SQL Delete performed on fields from **TABLE SQL** table. This action is specified by the value of the second API call parameter **VALUE2**.

```
{
    "info":
   {
       "version":"5.0.4.1"
   },
    "extensions":
    [
        {
           "apiKey":"a1",
           "name":"API2",
           "type": "call",
           "parameters":
           [  {
                "position":1,
                "label": "p1",
                "resolve": true,
                "locator":{
                "offset":1,
                "length":9
                  }
                  },
                  {
                "position":2,
                "label": "p2",
                "resolve": true,
                "locator":{
                "offset":1,
                "length":9
                  }
                  },
                  {
                "position":2,
                "label": "p3",
                "resolve": false
                  },
                  {
                "position":2,
                "label": "p4",
                "resolve": false
                  }
           ]
        }
    ]
}
```

The `ue-config.json` file specifies the type of user exit file (for v5.0.4, only `.Json` file is supported) and the path where the user exit file is located.

```
{
  "schemaVersion":"1.0.1",
  "documentVersion":"1.0.1",
  "ueConfig" :[
   {"name": "uejson",
    "type":"file-json",
    "location":"location from disk/resolution.json",
    "appliesTo":["a1"]
   }
]
}
```

The `resolution.json` file specifies:

- if **VALUE1** is found in parameter values the API2 call will become a SQL Select on *COL11* and *COL12* fields from **TABLE1 SQL** Table.
- if **VALUE2** is found in parameter values the API2 call will become a SQL Delete on *COL21* field from **TABLE2 SQL** Table.

```
{
    "schemaVersion": "1.0.1",
    "documentVersion": "1.0.1",
    "entries": [{
        "input": {
                    "apiKey": "a1",
          "apiText": "API2",
          "params": [
        {
          "value": "VALUE1"
        },
        {
          "paramKey": "p2",
          "paramMode": "byRef"
        },
        {
          "paramKey": "p3",
          "paramMode": "byRef"
        },
        {
          "paramKey": "p4",
          "paramMode": "byRef"
        }
      ]
      },
     "resolutions": [
       {
         "action": "SQLSelect",
         "resources": [
           {
             "resourceType": "SQLTable",
             "value": "TABLE1",
             "read": true
           }
         ],
         "clauses": [
           {
             "clause": "into",
             "from":
               {
                  "resourceType": "SQLField",
                  "qualifiers": [
                    "TABLE1"
                  ],
                  "value": "COL11",
                  "read": true
               },
              "to": {
                  "resourceType": "Variable",
                  "paramKey": "p2",
                  "paramMode": "byRef",
                  "read": false
               }
         },
         {
```

```
              "clause": "into",
              "from":
                {
                  "resourceType": "SQLField",
                  "qualifiers": [
                    "TABLE1"
                  ],
                  "value": "COL12",
                 "read": true
                },
               "to": {
                  "resourceType": "Variable",
                  "paramKey": "p3",
                  "paramMode": "byRef",
                  "read": false
                }
            },
            {
              "clause": "where",
              "from":
                {
                  "resourceType": "SQLField",
                  "qualifiers": [
                    "TABLE1"
                  ],
                  "value": "COL11",
                  "read": false
                },
                "to":{
                  "resourceType": "Variable",
                  "paramKey": "p4",
                  "paramMode": "byRef",
                  "read": true
                }
            }
          ]
    }]
},{
        "input": {
            "apiKey": "a1",
            "apiText": "API2",
            "params": [
          {
            "value": "VALUE2"
          },
          {
            "paramKey": "p2",
            "paramMode": "byRef"
          },
          {
            "paramKey": "p3",
            "paramMode": "byRef"
          },
          {
            "paramKey": "p4",
            "paramMode": "byRef"
          }
        ]
        },
      "resolutions": [
        {
          "action": "SQLDelete",
          "resources": [
            {
              "resourceType": "SQLTable",
              "value": "TABLE2",
      "read": true
            }
          ],
          "clauses": [
            {
              "clause": "where",
              "from":
                {
                  "resourceType": "SQLField",
                  "qualifiers": [
                    "TABLE2"
                  ],
                  "value": "COL21",
                  "read": false
                },
                "to":{
```

```
                    "resourceType": "Variable",
                    "paramKey": "p4",
                    "paramMode": "byRef",
                    "read": false
                 }
        }]
     }]
}]
}.
```

The following *json* schemas might help you create the configurations files.

Schema for `API_Config.json`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "AD Extensebility extensions definition file",
    "type": "object",
    "properties": {
        "info": {
            "type": "object",
            "properties": {
                "version": {
                    "type": "string",
                    "description": "The extension file version",
                    "pattern": "^[0-9]+\\.[0-9]+\\.[0-9]+"
                }
            },
            "required": [
                "version"
            ]
        },
        "extensions": {
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Extension"
            }
        }
    },
    "required": [
        "info",
        "extensions"
    ],
    "definitions": {
        "Extension": {
            "oneOf": [
                {
                    "$ref": "#/definitions/ExtensionType1"
                },
                {
                    "$ref": "#/definitions/ExtensionType2"
                },
                {
                    "$ref": "#/definitions/ExtensionType3"
                },
            {
                    "$ref": "#/definitions/ExtensionType4"
                }
            ]
        },
        "LocatorType1": {
            "type": "object",
            "description": "Locator specifying the parameter value from a record's child",
            "properties": {
                "name": {
                    "type": "string",
                    "description": "Name of the record child from which to take the parameter value",
                    "minLength": 1
                }
            },
            "required": [
                "name"
            ],
            "additionalProperties": false
        },
        "LocatorType2": {
            "type": "object",
            "description": "Locator specifying the parameter value by offset and length",
            "properties": {
                "offset": {
                    "type": "integer",
                    "description": "Starting offset for the value",
                    "minimum": 0
                },
                "length": {
                    "type": "integer",
                    "description": "Length of the value to be read",
                    "minimum": 0
                }
            },
            "required": [
                "offset",
                "length"
            ],
            "additionalProperties": false
        },
        "ExtensionType1": {
            "type": "object",
            "properties": {
                "apiKey": {
                    "type": "string",
                    "minLength": 1
```

```
            },
            "name": {
                "type": "string",
                "minLength": 1
            },
            "type": {
                "type": "string",
                "enum": [
                    "call"
                ],
                "description": "Extension for call statements",
                "minLength": 1
            },
            "parameters": {
                "type": "array",
                "minItems": 1,
                "items": {
                    "$ref": "#/definitions/ParameterType1"
                }
            }
        },
        "required": [
            "apiKey",
            "name",
            "type",
            "parameters"
        ],
        "additionalProperties": false
    },
    "ExtensionType2": {
      "type": "object",
      "properties": {
        "apiKey": {
          "type": "string",
          "minLength": 1
        },
        "name": {
          "type": "string",
          "minLength": 1
        },
        "type": {
          "type": "string",
          "enum": [
            "cics"
          ],
          "description": "Extension for cics statements",
          "minLength": 1
        },
        "parameters": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref": "#/definitions/ParameterType2"
          }
        }
      },
      "required": [
        "apiKey",
        "name",
        "type",
        "parameters"
      ],
      "additionalProperties": false
    },
     "ExtensionType3": {
      "type": "object",
      "properties":
      {
        "apiKey":
        {
          "type": "string",
          "minLength": 1
        },
        "name":
        {
          "type": "string",
          "minLength": 1
        },
        "type": {
          "type": "string",
          "enum": [
            "JclPgmCall"
          ],
          "description": "Extension for JCL call statements",
          "minLength": 1
        },
        "parameters": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref": "#/definitions/ParameterType3"
          }
        }
      },
        "required": [
            "apiKey",
            "name",
            "type",
            "parameters"
        ],
        "additionalProperties": false
      },
    "ExtensionType4": {
      "type": "object",
      "properties":
      {
        "apiKey":
        {
          "type": "string",
```

```
            "minLength": 1
          },
          "name":
          {
            "type": "string",
            "minLength": 1
          },
          "type": {
            "type": "string",
            "enum": [
              "JclPgmCallParm"
            ],
            "description": "Extension for JCL call statements with PARM",
            "minLength": 1
          },
          "parameters": {
            "type": "array",
            "minItems": 1,
            "items": {
              "$ref": "#/definitions/ParameterType4"
            }
          }
        },
          "required": [
              "apiKey",
              "name",
              "type",
              "parameters"
          ],
          "additionalProperties": false
        },
      "ParameterType1": {
          "type": "object",
          "description": "Parameter for call statements",
          "properties": {
              "position": {
                  "type": "integer",
                  "description": "Position of the parameter in the call",
                  "minimum": 1
              },
              "label": {
                  "type": "string",
                  "description": "Identifier for the parameter",
                  "minLength": 1
              },
              "resolve": {
                  "type": "boolean",
                  "description": "Specify whether to determine the values of the parameter or not",
                  "default": true
              },
              "optional": {
                  "type": "boolean",
                  "description": "Specify whether the parameter is optional or not",
                  "default": false
              },
              "locator": {
                  "oneOf": [
                      {
                          "$ref": "#/definitions/LocatorType1"
                      },
                      {
                          "$ref": "#/definitions/LocatorType2"
                      }
                  ]
              }
          },
          "required": [
              "position",
              "label",
              "resolve"
          ],
          "additionalProperties": false
      },
      "ParameterType2": {
          "type": "object",
          "description": "Parameter for CICS statements",
          "properties": {
              "name": {
                  "type": "string",
                  "description": "Name of the parameter in the CICS statement",
                  "minLength": 1
              },
              "label": {
                  "type": "string",
                  "description": "Identifier for the parameter",
                  "minLength": 1
              },
              "resolve": {
                  "type": "boolean",
                  "description": "Specify whether to determine the values of the parameter or not",
                  "default": true
              },
              "optional": {
                  "type": "boolean",
                  "description": "Specify whether the parameter is optional or not",
                  "default": false
              },
              "locator": {
                  "oneOf": [
                      {
                          "$ref": "#/definitions/LocatorType1"
                      },
                      {
                          "$ref": "#/definitions/LocatorType2"
                      }
                  ]
              }
          },
          "required": [
```

```
                "name",
                "label",
                "resolve"
            ],
            "additionalProperties": false
        },

        "ParameterType3": {
            "type": "object",
            "description": "Parameter for JCL calls",
            "properties": {
              "name":
              {
                "type": "string",
                "description": "Name of the parameter in the JCL call",
                "minLength": 1
              },
              "label":
              {
                "type": "string",
                "description": "Identifier for the parameter",
                "minLength": 1
              },
             "optional": {
                    "type": "boolean",
                    "description": "Specify whether the parameter is optional or not",
                    "default": false
             }
            },
            "required": [
                "name",
                "label"
            ],
            "additionalProperties": false
        },
      "ParameterType4": {
            "type": "object",
            "description": "Parameter for JCL calls with PARM",
            "properties": {
              "position":
              {
                "type": "integer",
                "description": "Position of the parameter in the PARM attribute of the JCL call",
                "minLength": 1
              },
              "label":
              {
                "type": "string",
                "description": "Identifier for the parameter",
                "minLength": 1
              },
             "optional": {
                    "type": "boolean",
                    "description": "Specify whether the parameter is optional or not",
                    "default": false
             }
            },
            "required": [
                "position",
                "label"
            ],
            "additionalProperties": false
        }
    }
}
```

Schema for `ue-config.json`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "version": "1.0.1",
    "type": "object",
    "description": "An ordered list of user exits",
    "properties": {
        "schemaVersion": {
            "description": "This json schema version",
            "$ref": "#/definitions/versionDef"
        },
        "documentVersion": {
            "description": "The user exit json file version",
            "$ref": "#/definitions/versionDef"
        },
        "ueConfig": {
            "type": "array",
            "description": "An array which contains user exit configurations",
            "items": {
                "type": "object",
                "description": "An user exit configuration object.",
                "properties": {
                    "name": {
                        "type": "string",
                        "description": "A name for this user exit configuration."
                    },
                    "type": {
                        "type": "string",
                        "description": "The user exit type",
                        "enum": [
                            "file-json",
                            "utility",
                            "dependency-file-json",
                            "dependency-utility",
                            "jcl-file-json",
                            "jcl-utility"
                        ]
                    },
                    "location": {
                        "type": "string",
                        "description": "The path of the file containing the resolutions (applies for file-json type only)."
                    },
                    "appliesTo": {
                        "type": "array",
                        "description": "An array of API (defined in API config json) to which this user-exit is applied.",
                        "items": {
                            "type": "string",
                            "description": "An extensibility API key as defined in the API config json file."
```

```
                        },
                        "minItems": 1,
                        "uniqueItems": true
                    },
                    "command": {
                        "type": "array",
                        "description": "An array of strings used to call an external utility in order to obtain the user exit resolutions.",
                        "items": {
                            "type": "string"
                        }
                    }
                },
                "required": [
                    "name",
                    "type",
                    "appliesTo"
                ],
                "uniqueItems": true
            }
        }
    },
    "required": [
        "ueConfig",
        "schemaVersion",
        "documentVersion"
    ],
    "definitions": {
        "versionDef": {
            "type": "string",
            "description": "Schema/Document versioning pattern",
            "pattern": "^[0-9]{1,3}\\.[0-9]{1,3}(\\.[0-9]{1,3})?$"
        }
    }
}
```

Schema for `resolution.json`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "version": "1.0.2",
    "type": "object",
    "description": "The json schema for a file-json based AD resolutions.",
    "properties": {
        "schemaVersion": {
            "description": "This json schema version.",
            "$ref": "#/definitions/versionDef"
        },
        "documentVersion": {
            "description": "The user exit json file version.",
            "$ref": "#/definitions/versionDef"
        },
        "entries": {
            "type": "array",
            "description": "An array containing the user exit content.",
            "items": {
                "type": "object",
                "description": "A user exit content object.",
                "properties": {
                    "resolutions": {
                        "type": "array",
                        "description": "An array which contains the user exit resolution objects.",
                        "items": {
                            "anyOf": [
                                {
                                    "$ref": "#/definitions/localResolutionDef"
                                },
                                {
                                    "$ref": "#/definitions/externalResolutionDef"
                                },
                                {
                                    "$ref": "#/definitions/callLocalResolutionDef"
                                },
                                {
                                    "$ref": "#/definitions/callExternalResolutionDef"
                                }
                            ]
                        }
                    },
                    "input": {
                        "type": "object",
                        "description": "An object which contains the inputs for the user exit.",
                        "properties": {
                            "apiKey": {
                                "type": "string",
                                "description": "The parameter key as defined in the API config json."
                            },
                            "apiText": {
                                "type": "string",
                                "description": "The name of the API program as defined in the API config json."
                            },
                            "params": {
                                "type": "array",
                                "description": "An array of 'value' objects definitions. The number and the order of the items MUST be the same with the
number and the order of the parameters defined in the API config json",
                                "items": {
                                    "anyOf": [
                                        {
                                            "$ref": "#/definitions/paramValue"
                                        },
                                        {
                                            "$ref": "#/definitions/directValue"
                                        },
                                        {
                                            "$ref": "#/definitions/optionalValue"
                                        }
                                    ]
                                }
                            }
                        },
                        "required": [
                            "apiKey",
                            "params"
                        ],
                        "additionalProperties": false
                    }
                },
                "additionalProperties": false,
                "required": [
                    "resolutions",
                    "input"
                ]
            }
        }
    },
    "required": [
        "entries",
        "schemaVersion",
        "documentVersion"
```

```
    ],
    "definitions": {
        "versionDef": {
            "type": "string",
            "description": "A schema/document versioning pattern.",
            "pattern": "^[0-9]{1,3}\\.[0-9]{1,3}(\\.[0-9]{1,3})?$"
        },
        "externalActionsDef": {
            "description": "",
            "type": "string",
            "enum": [
                "CICSStart",
                "CICSReturn",
                "IMSStartTransaction"
            ]
        },
        "localActionsDef": {
            "description": "",
            "type": "string",
            "enum": [
                "CICSStart",
                "CICSReturn",
                "SQLSelect",
                "SQLInsert",
                "SQLDelete",
                "SQLUpdate",
                "FileRead",
                "FileWrite",
                "FileRewrite",
                "FileDelete",
                "FileSort",
                "CICSFileRead",
                "CICSFileWrite",
                "CICSFileRewrite",
                "CICSFileDelete",
                "CICSSendMap",
                "CICSReceiveMap",
                "IMSSendMap",
                "IMSReceiveMap",
                "IMSStartTransaction",
                "IMSDBRead",
                "IMSDBReadNext",
                "IMSDBInsert",
                "IMSDBDelete",
                "IMSDBUpdate"
            ]
        },
        "paramValue": {
            "type": "object",
            "description": "A generic 'value' object which allows the user exit to define values for the resolution or the input values with references
to the paramKey.",
            "properties": {
                "paramKey": {
                    "type": "string",
                    "description": "The parameter key as defined in the API config json file."
                },
                "paramMode": {
                    "type": "string",
                    "description": "A marker which defines how the parameter value is read by the resolver.",
                    "enum": [
                        "byValue",
                        "byRef"
                    ]
                }
            },
            "additionalProperties": false,
            "required": [
                "paramKey",
                "paramMode"
            ]
        },
        "directValue": {
            "type": "object",
            "description": "A generic 'value' object which allows the user exit to define values for the resolution or the input values without
refererring to a specific paramKey.",
            "properties": {
                "value": {
                    "type": "string",
                    "description": "The value for this 'value' object."
                }
            },
            "additionalProperties": false,
            "required": [
                "value"
            ]
        },
        "optionalValue": {
            "type": "object",
            "description": "A generic 'value' object which allows the user exit to define values for the input values which will be ignored by the AD
Resolver",
            "properties": {
                "ignore": {
                    "type": "boolean",
                    "description": "If true the 'value' object MUST be ignored by the AD Resolver."
                }
            },
            "additionalProperties": false,
            "required": [
                "ignore"
            ]
        },
        "resourceTypeDef": {
            "type": "object",
            "description": "An object which will contain a resource type definition for resolutions.",
            "properties": {
                "read": {
                    "type": "boolean",
                    "description": "A marker for the resource behavior (true: the resource is read, false - the resource is written.)"
                },
                "resourceType": {
                    "type": "string",
                    "description": "The resource type",
                    "enum": [
                        "CobolProgram",
                        "PL/1Program",
                        "AssemblerProgram",
                        "Variable",
                        "CICSTran",
                        "SQLTable",
                        "SQLField",
                        "File",
                        "CICSMap",
                        "IMSMessageDescriptor",
                        "IMSTransaction",
                        "IMSDBSegment"
                    ]
                },
                "paramKey": {
                    "type": "string",
                    "description": "The parameter key as defined in the API config json file."
                },
```

```
                "paramMode": {
                    "type": "string",
                    "description": "A marker which defines how the parameter value is read by the resolver.",
                    "enum": [
                        "byValue",
                        "byRef"
                    ]
                },
                "qualifiers": {
                    "type": "array",
                    "description": "The strings will be used to qualify the resource. It is required for resourceType SQLField, IMSMessageDescriptor,
IMSDBSegment",
                    "items": {
                        "type": "string",
                        "description": "A string qualifier."
                    }
                },
                "value": {
                    "type": "string",
                    "description": "A string value for the resource type."
                },
                "annText": {
                        "type": "string",
                        "description": "A string which defines the text of the annotation."
                },
                "annKeyword": {
                        "type": "string",
                        "description": "A string which defines the key of the annotation."
                }
            },
            "additionalProperties": false
        },
        "clausesDef": {
            "type": "array",
            "description": "An array containing SQL statements related options.",
            "items": {
                "type": "object",
                "description": "",
                "properties": {
                    "clause": {
                        "type": "string",
                        "description": "The relationship between the 'from' and the 'to' 'value' objects.",
                        "enum": [
                            "into",
                            "where",
                            "set"
                        ]
                    },
                    "from": {
                        "description": "A SQLField value object",
                        "$ref": "#/definitions/resourceTypeDef"
                    },
                    "to": {
                        "description": "A host variable 'value' object.",
                        "$ref": "#/definitions/resourceTypeDef"
                    }
                },
                "additionalProperties": false
            }
        },
        "targetDef": {
            "description": "The targeted resource (if any). Will be deprecated in the next versions.",
            "$ref": "#/definitions/resourceTypeDef"
        },
        "localResolutionDef": {
            "properties": {
                "action": {
                    "$ref": "#/definitions/localActionsDef"
                },
                "resources": {
                    "type": "array",
                    "description": "An array containing the user exit values for the given resolution.",
                    "items": {
                        "description": "",
                        "$ref": "#/definitions/resourceTypeDef"
                    }
                },
                "clauses": {
                    "$ref": "#/definitions/clausesDef"
                }
            },
            "additionalProperties": false,
            "required": [
                "action"
            ]
        },
        "externalResolutionDef": {
            "properties": {
                "action": {
                    "$ref": "#/definitions/externalActionsDef"
                },
                "location": {
                    "type": "string",
                    "description": "A string which defines the location (another AD project or a CICS region) where the API action is performed."
                },
                "locationType": {
                    "type": "string",
                    "description": "The AD project name or a CICS region.",
                    "enum": [
                        "application",
                        "CICSRegion"
                    ]
                },
                "resources": {
                    "type": "array",
                    "description": "An array containing the user exit values for the given resolution.",
                    "items": {
                        "description": "",
                        "$ref": "#/definitions/resourceTypeDef"
                    }
                }
            },
            "additionalProperties": false,
            "required": [
                "action", "location", "locationType"
            ]
        },
        "callLocalResolutionDef": {
            "properties": {
                "action": {
                    "enum": [
                        "call"
                    ]
                },
                "target": {
                    "$ref": "#/definitions/targetDef"
                },
                "resources": {
                    "type": "array",
                    "description": "An array containing the user exit values for the given resolution.",
                    "items": {
```

```
                    "description": "",
                    "$ref": "#/definitions/resourceTypeDef"
                }
            }
        },
        "additionalProperties": false,
        "required": [
            "action",
            "target"
        ]
    },
    "callExternalResolutionDef": {
        "properties": {
            "action": {
                "enum": [
                    "call"
                ]
            },
            "location": {
                "type": "string",
                "description": "A string which defines the location (another AD project or a CICS region) where the API action is performed."
            },
            "locationType": {
                "type": "string",
                "description": "The AD project name or a CICS region.",
                "enum": [
                    "application",
                    "CICSRegion"
                ]
            },
            "target": {
                "$ref": "#/definitions/targetDef"
            },
            "resources": {
                "type": "array",
                "description": "An array containing the user exit values for the given resolution.",
                "items": {
                    "description": "",
                    "$ref": "#/definitions/resourceTypeDef"
                }
            }
        },
        "additionalProperties": false,
        "required": [
            "action",
            "location",
            "locationType",
            "target"
        ]
    }
}
```

JCL Call Extensibility Examples

JCL Call Extensibility Use Cases

The purpose of this chapter is to help users understand how the Configuration files must be created in order to integrate their business-specific cases in the analysis.

Case 1. When parameter value is received from DD cards

The purpose of this chapter is to help users understand how the Configuration files must be created in order to integrate their business-specific cases in the analysis.

This use case is created to cover the situation when a JCL Call to an API program is translated into calls to other programs, based on the parameter value received by the API program in DD cards. The following represents the **JOBSAMPLE** JCL job:

```
//JOBSAMPLE JOB  ,'SAMPLE JCL JOB',
//             CLASS=C,
//             MSGCLASS=P,
//             COND=(4,LT),
//             REGION=4096K
//STP010   EXEC  PGM=APIJCL
//DDOUT1   DD  DSN=PJ.ABC.DATASET1,
//             UNIT=(MTLIB,,DEFER),
//             DISP=(NEW,KEEP),
//             DCB=TRTCH=C,
//             LABEL=(,SL),
//             VOL=SER=DXXXXX
//APIPARAM    DD *
1111 ABC --TEST-- VALUE
//
```

The **JOBSAMPLE** JCL job contains a call to a specific API program named **APIJCL**.

To resolve the Call to "APIJCL", the `JCL_Config.json` file specifies that it is necessary to resolve the value of the **APIPARAM** parameter.

Note: This JSON contains as parameters the corresponding name of the JCL DD card used in the JCL Call.

```
{
  "info": {
    "version": "5.0.3"
  },
    "extensions":
    [
       {
          "apiKey":"a1",
          "name": "APIJCL",
          "type": "JclPgmCall",

           "parameters":
           [
            {
              "name": "APIPARAM",
              "label": "param1"
            }
           ]
        }

    ]
}
```

The `ue-config.json` file specifies the type of user exit file and the path where the user exit file for a certain API is located.

```
{
  "schemaVersion":"1.0.1",
  "documentVersion":"1.0.1",
  "ueConfig" :[
   {"name": "uejson",
    "type":"jcl-file-json",
    "location":"D:/EZSourceBuildProjects/API_JCL_PROJECTS/DOC_JCL_DEP/API Config/Resolutions/
resolutionJCL.json",
    "appliesTo":["a1"]
   }
]
}
```

The `resolutionJCL.json` file specifies that, if the value of the received parameter is *"1111 ABC--TEST--VALUE"*, the JCL Call to `"APIJCL"` is replaced with calls to **PROGRAM1** and **PROGRAM2** COBOL programs.

Note:

- The values of the parameters present in the `resolutionJCL.json` file must be identical to the values of the corresponding DD cards in the JCL, considering a maximum of 71 characters on the line.
- The values of the DD cards can come either inline `DD *` or from a controlled file `DD DSN=<file path>`.

The following annotations are present in the `resolutionJCL.json` file:

- For **PROGRAM1** - annotation `"annText": "ANNOTATION1"` and annotation keyword `"annKeyword" : "API_RESOLUTION"`.
- For **PROGRAM2** - annotation `"annText": "ANNOTATION2"` and annotation keyword `"annKeyword" : "API_RESOLUTION"`.

```
{
    "schemaVersion": "1.0",
    "documentVersion": "1.0",
    "entries": [{
        "input": {
            "apiKey": "a1",
            "apiText": "APIJCL",
            "params": [{
                    "value": "1111 ABC--TEST--VALUE"
                }

            ]
        },
        "resolutions": [{
                "action": "call",
                "target": {
```

```
                    "resourceType": "CobolProgram",
                    "value": "PROGRAM1",
                    "read": true
                },
                "annText": "ANNOTATION1",
                "annKeyword" : "API_RESOLUTION"
            },
            {
                "action": "call",
                "target": {
                    "resourceType": "CobolProgram",
                    "value": "PROGRAM2",
                    "read": true
            },
                "annText": "ANNOTATION2",
                "annKeyword" : "API_RESOLUTION"
            }]
    }
    ]
}
```

Case 2. When parameter value is received from PARM attribute

This use case is created to cover the situation when a JCL Call to an API program is translated into calls to other programs, based on the parameter value received by the API program in the PARM attribute. The following represents the JOBSAMPLE JCL job:

```
//JOBSAMPLE JOB  ,'SAMPLE JCL JOB',
//             CLASS=C,
//             MSGCLASS=P,
//             COND=(4,LT),
//             REGION=4096K
//STP040   EXEC PGM=APIJCL2,REGION=0M,
//         PARM='COBOLPGM1,100,1'
//SYSUT1   DD  DSN=ABC.TST,
//             DISP=(OLD,KEEP)
```

The **JOBSAMPLE** JCL job contains a call to a specific API program named **APIJCL2**.

To resolve the Call to "APIJCL2", the JCL_Config.json file specifies that it is necessary to resolve the value of the parameter in the first position from the **PARM** attribute of the **EXEC** statement.

```
{
  "info": {
    "version": "5.0.3"
  },
    "extensions":
    [
       {
          "apiKey":"a1",
          "name": "APIJCL2",
          "type": "JclPgmCallParm",

            "parameters":
           [
            {
             "position":1,
              "label": "param1"
            }
           ]
        }

    ]
}
```

The ue-config.json file specifies the type of user exit file and the path where the user exit file for a certain API is located.

```
{
  "schemaVersion":"1.0.1",
  "documentVersion":"1.0.1",
  "ueConfig" :[
   {"name": "uejson",
    "type":"jcl-file-json",
    "location":"D:/EZSourceBuildProjects/API_JCL_PROJECTS/DOC_JCL_DEP/API Config/Resolutions/
```

```
resolutionJCL.json",
    "appliesTo":["a1"]
   }
]
}
```

The `resolutionJCL.json` file specifies that, if the value of the received parameter is *COBOLPGM1*, the JCL Call to `"APIJCL2"` is replaced with calls to **COBOLPGM1** COBOL program.

Note: The values of the parameters present in the `resolutionJCL.json` file must be identical to the value of the corresponding parameter in the **PARM** attribute.

For **COBOLPGM1**, the `"annText": "JCL1PARM"` annotation and `"annKeyword" : "activityJCL"` annotation keyword are present in the `resolutionJCL.json` file:

```
{
    "schemaVersion": "1.0",
    "documentVersion": "1.0",
    "entries": [{
        "input": {
            "apiKey": "a1",
            "apiText": "APIJCL2",
            "params": [{
                    "value":"COBOLPGM1"
                }
            ]
        },
        "resolutions": [{
                "action": "call",
                "target": {
                    "resourceType": "CobolProgram",
                    "value": "COBOLPGM1",
                    "read": true,
                "annText": "JCL1PARM",
                "annKeyword" : "activityJCL"
            }}]
    }
    ]
}
```

JSON Schemas

The `API_Config.json` schema, present in “API/Macro Call Extensibility Examples ” on page 500 introduces a new API type `"type":"JclPgmCall"`.

The `ue-config.json` schema, present in “API/Macro Call Extensibility Examples ” on page 500 introduces a new API type `"type":"jcl-file-json"`.

Schema for JCL `resolutionJCL.json`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "version": "1.0.0",
    "type": "object",
    "description": "The json schema for a jcl-json based AD resolutions.",
    "properties": {
        "schemaVersion": {
            "description": "This json schema version.",
            "$ref": "#/definitions/versionDef"
        },
        "documentVersion": {
            "description": "The user exit json file version.",
            "$ref": "#/definitions/versionDef"
        },
        "entries": {
            "type": "array",
            "description": "An array containing the user exit content.",
            "items": {
                "type": "object",
                "description": "A user exit content object.",
                "properties": {
                    "resolutions": {
                        "type": "array",
                        "description": "An array which contains the user exit resolution objects.",
                        "items": { "$ref": "#/definitions/callLocalResolutionDef" }
                    },
                    "input": {
                        "type": "object",
                        "description": "An object which contains the inputs for the user exit.",
                        "properties": {
                            "apiKey": {
                                "type": "string",
                                "description": "The parameter key as defined in the JCLAPI config json."
                            },
                            "apiText": {
                                "type": "string",
                                "description": "The name of the API program as defined in the JCLAPI config json."
                            },
                            "params": {
                                "type": "array",
                                "description": "An array of 'value' objects definitions. The number and the order of the items MUST be the same with the
```

```
number and the order of the parameters defined in the API config json",
                                "items": {
                                    "anyOf": [
                                        {
                                            "$ref": "#/definitions/paramValue"
                                        },
                                        {
                                            "$ref": "#/definitions/directValue"
                                        }
                                    ]
                                }
                            }
                        },
                        "required": [
                            "apiKey",
                            "params"
                        ],
                        "additionalProperties": false
                    }
                },
                "additionalProperties": false,
                "required": [
                    "resolutions",
                    "input"
                ]
            }
        }
    },
    "required": [
        "entries",
        "schemaVersion",
        "documentVersion"
    ],
    "definitions": {
        "versionDef": {
            "type": "string",
            "description": "A schema/document versioning pattern.",
            "pattern": "^[0-9]{1,3}\\.[0-9]{1,3}(\\.[0-9]{1,3})?$"
        },
        "paramValue": {
            "type": "object",
            "description": "A generic 'value' object which allows the user exit to define values for the resolution or the input values with references
to the paramKey.",
            "properties": {
                "paramKey": {
                    "type": "string",
                    "description": "The parameter key as defined in the API config json file."
                },
                "paramMode": {
                    "type": "string",
                    "description": "A marker which defines how the parameter value is read by the resolver.",
                    "enum": [
                        "byValue",
                        "byRef"
                    ]
                }
            },
            "additionalProperties": false,
            "required": [
                "paramKey",
                "paramMode"
            ]
        },
        "directValue": {
            "type": "object",
            "description": "A generic 'value' object which allows the user exit to define values for the resolution or the input values without
refererring to a specific paramKey.",
            "properties": {
                "value": {
                    "type": "string",
                    "description": "The value for this 'value' object."
                }
            },
            "additionalProperties": false,
            "required": [
                "value"
            ]
        },
        "resourceTypeDef": {
            "type": "object",
            "description": "An object which will contain a resource type definition for resolutions.",
            "properties": {
                "read": {
                    "type": "boolean",
                    "description": "A marker for the resource behavior (true: the resource is read, false - the resource is written.)"
                },
                "resourceType": {
                    "type": "string",
                    "description": "The resource type",
                    "enum": [
                        "CobolProgram",
                        "PL/1Program",
                        "AssemblerProgram"
                    ]
                },
                "paramKey": {
                    "type": "string",
                    "description": "The parameter key as defined in the JCLAPI config json file."
                },
                "paramMode": {
                    "type": "string",
                    "description": "A marker which defines how the parameter value is read by the resolver.",
                    "enum": [
                        "byValue",
                        "byRef"
                    ]
                },
                "qualifiers": {
                    "type": "array",
                    "description": "An array of strings. The strings will be used to qualify the resource.",
                    "items": {
                        "type": "string",
                        "description": "A string qualifier."
                    }
                },
                "value": {
                    "type": "string",
                    "description": "A string value for the resource type."
                },
                "annText": {
                        "type": "string",
                        "description": "A string which defines the key of the annotation."
                },
                "annKeyword": {
                        "type": "string",
                        "description": "A string which defines the text of the annotation."
                }
            },
            "additionalProperties": false
        },
        "targetDef": {
```

```
            "description": "The targeted resource (if any). Will be deprecated in the next versions.",
            "$ref": "#/definitions/resourceTypeDef"
        },
        "callLocalResolutionDef": {
            "properties": {
               "action": {
                    "enum": [
                        "call"
                    ]
                },
                "target": {
                    "$ref": "#/definitions/targetDef"
                }
            },
            "additionalProperties": false,
            "required": [
                "action",
                "target"
            ]
        }
    }
}
```

Dependency Extensibility Examples

Dependency Extensibility Use Case

The purpose of this use case is to help users understand how the Configuration files must be created to integrate their business-specific cases in the analysis.

This use case is created to cover the situation when a user creates mappings between transactions and programs.

The `API_Dependency.json` file specifies the APIs that return mappings between resources.

Note: Each API has a name and label. The APIs for dependencies have the attribute `"type":"post_build".`

```
{
    "info":
   {
      "version":"5.1.0"
   },
    "extensions":
    [
        {
          "apiKey":"a2",
          "name":"MAPPING_TRAN",
          "type":"post_build"
        }
    ]
}
```

The `ue-config.json` file specifies the type of user exit file and the path where the user exit file for a certain dependency API is located.

```
{
  "schemaVersion":"1.0.1",
  "documentVersion":"1.0.1",
  "ueConfig" :[
   {"name": "uejson",
    "type":"dependency-file-json",
    "location":"D:/EZSourceBuildProjects/API_JCL_PROJECTS/DOC_JCL_DEP/API Config/Resolutions/
resolutionDEP.json",
    "appliesTo":["a2"]
   }
]
}
```

The `resolutionDEP.json` file contains the mapping of resources that are returned by the `"MAPPING_TRAN"` API. In this specific case the mappings are as follows:

- Transaction `"TRAN1"` to program `"PROGRAM1"`
- Transaction `"TRAN2"` to program `"PROGRAM2"`

The following annotations are present in the `resolutionDEP.json` file:

- For **PROGRAM1** - annotation `"annText": "ANNOTATION3"` and annotation keyword `"annKeyword" : "API_RESOLUTION"`.

- For **PROGRAM2** - annotation `"annText": "ANNOTATION4"` and annotation keyword `"annKeyword" : "API_RESOLUTION"`.

The AD analysis shows the links between the mentioned resources.

```
{
    "schemaVersion": "1.0.0",
    "documentVersion": "1.0.0",
    "entries": [{
        "input": {
            "apiKey": "a2",
            "apiText": "MAPPING_TRAN"
        },
        "resolutions": [{
            "action": "dependency",
            "source": {
                "resourceType": "GenericTransaction",
                "value": "TRAN1"
            },
            "target": {
                "resourceType": "CobolProgram",
                "value": "PROGRAM1",
                "annText": "ANNOTATION3",
                "annKeyword" : "API_RESOLUTION"
            }
        },{
            "action": "dependency",
            "source": {
                "resourceType": "GenericTransaction",
                "value": "TRAN2"
            },
            "target": {
                "resourceType": "CobolProgram",
                "value": "PROGRAM2",
                "annText": "ANNOTATION4",
                "annKeyword" : "API_RESOLUTION"
            }
        } ]
    }]
}
```

JSON Schemas

Schema for Dependency `API_Dependency.json`

A new configuration file is introduced. The API for dependency has a new type of attribute `"type": "post_build"`.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "AD Extensibility dependencies definition file",
    "type": "object",
    "properties": {
        "info": {
            "type": "object",
            "properties": {
                "version": {
                    "type": "string",
                    "description": "The extension file version",
                    "pattern": "^[0-9]+\\.[0-9]+\\.[0-9]+"
                }
            },
            "required": [
                "version"
            ]
        },
        "extensions": {
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Extension"
            }
        }
    },
    "required": [
        "info",
        "extensions"
    ],
    "definitions": {
      "Extension":
      {
        "type": "object",
        "properties": {
          "apiKey": {
            "type": "string",
            "minLength": 1
          },
          "name": {
            "type": "string",
            "minLength": 1
          },
          "type": {
            "type": "string",
            "enum": [
              "post_build"
```

```
            ],
            "description": "Extension for dependencies",
            "minLength": 1
          }
        },
        "required": [
          "apiKey",
          "name",
          "type"
        ],
        "additionalProperties": false
      }
    }
}
}
```

The `ue-config.json` schema, present in the “API/Macro Call Extensibility Examples ” on page 500 introduces a new API type `"type":"dependency-file-json"`.

Schema for Dependency `resolution.json`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "version": "1.0.0",
    "type": "object",
    "description": "The json schema for a file-json based AD dependency resolutions.",
    "properties": {
        "schemaVersion": {
            "description": "This json schema version.",
            "$ref": "#/definitions/versionDef"
        },
        "documentVersion": {
            "description": "The user exit dependency json file version.",
            "$ref": "#/definitions/versionDef"
        },
        "entries": {
            "type": "array",
            "description": "An array containing the user exit dependency content.",
            "items": {
                "type": "object",
                "description": "A user exit dependency content object.",
                "properties": {
                    "input": {
                        "type": "object",
                        "description": "An object which contains the inputs for the user exit dependency.",
                        "properties": {
                            "apiKey": {
                                "type": "string",
                                "description": "The parameter key as defined in the API config json."
                            },
                            "apiText": {
                                "type": "string",
                                "description": "The name of the API_NOT_SOURCE as defined in the API config json."
                            }
                        },
                        "required": [
                            "apiKey"
                        ],
                        "additionalProperties": false
                    },
                    "resolutions": {
                        "type": "array",
                        "description": "An array which contains the user exit resolution objects.",
                        "properties": {
                            "action": {
                                "type": "string",
                                "description" : "The name of the action, expect dependency.",
                                "enum": [
                                    "dependency"
                                ]
                            },
                            "source": {
                                "$ref": "#/definitions/sourceDef"
                            },
                            "target": {
                                "$ref": "#/definitions/targetDef"
                            }
                        }
                    }
                },
                "additionalProperties": false,
                "required": [
                    "resolutions",
                    "input"
                ]
            }
        }
    },
    "required": [
        "entries",
        "schemaVersion",
        "documentVersion"
    ],
    "definitions": {
        "versionDef": {
            "type": "string",
            "description": "A schema/document versioning pattern.",
            "pattern": "^[0-9]{1,3}\\.[0-9]{1,3}(\\.[0-9]{1,3})?$"
        },
        "sourceDef": {
            "properties": {
                "resourceType": {
                        "type": "string",
                        "description": "A string which defines the source type of the resource from depedency relation."
                },
                "value": {
                        "type": "string",
                        "description": "A string which defines the source name of the resource from depedency relation."
                },
                "qualifier": {
                    "type": "string",
                    "description": "A string which defines the qualifier of the resource from depedency relation."
                }
            },
            "required": [
                "resourceType",
                "value"
            ]
        },
        "targetDef": {
            "properties": {
                "resourceType": {
                        "type": "string",
                        "description": "A string which defines the target type of the resource from depedency relation."
                },
```

```
                "value": {
                        "type": "string",
                        "description": "A string which defines the target name of the resource from depedency relation."
                },
                "annText": {
                        "type": "string",
                        "description": "A string which defines the key of the annotation."
                },
                "annKeyword": {
                        "type": "string",
                        "description": "A string which defines the text of the annotation."
                }
            }
        }
    }
}
```

IBM AD REST API Example

The base path for the **Application Discovery REST API** is `/ws`. Also for all sample requests, add `/ws` after **`{host}:{port}`**:

```
curl--userusername:passwordhttps://{host}:{port}/ws/projects
```

GET /projects

Description: Get all projects.

Sample request:

```
curl --user username:password https://{host}:{port}/projects
```

Sample response:

```
{
    "projects": [
        {
            "name": "zMobile",
            "type": "mainframe",
            "uuid": "zMobile",
            "defined": 1511983789
        },
        {
            "name": "SAMPLE-PROJECT",
            "type": "mainframe",
            "uuid": "SAMPLE-PROJECT",
            "defined": 1511983787
        },
        {
            "name": "JKEBank",
            "type": "mainframe",
            "uuid": "JKEBank",
            "defined": 1511983790
        }
    ]
}
```

Where:

- **`projects`**: representing an array containing all the project entities. The project entity structure:
  - **`name`**: representing the name of AD project.
  - **`type`**: representing the type of the AD project (only *mainframe* at the moment).
  - **`uuid`**: representing the UUID of the AD project (the AD project name at this moment, it will be replaced with UUID variant 4).
  - **`defined`**: representing the UNIX timestamp (in seconds) when the project was added to the collection.

GET /projects/{project-uuid}/datasets

Where *{project-uuid}* is a unique identifier for a project.

Description: Get all datasets available in an AD project.

Sample request:

```
curl --user username:password https://{host}:{port}/projects/SAMPLE-PROJECT/datasets
```

Sample response:

```
{
    "meta": {
        "project": {
            "name": "SAMPLE-PROJECT",
            "uuid": "SAMPLE-PROJECT",
        },
        "snapshot": 1502354654
    },
    "data": [
        {
            "uid": "01da56d83ec1c04be901ebe162957e3d3eaf6e49a85a0f31497f535b1f567b32",
            "name": "CB12V51.GENAPP.KSDSCUST.REPORT",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "(,,,4)",
                "datasetType": "GDG"
            }
        },
        {
            "uid": "09c2ef33f622e62da876e1d73a7e9e8f3aa8236034973a3f232acbf4da8509ee",
            "name": "DFH520.CICS.SDFJAUTH",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "DATASET"
            }
        },
        {
            "uid": "0c296329019f68889434bc14103de4475c3a680c64ee91f9a8f331c736b30257",
            "name": "DFH530.CNTL.CICS.DFHTEMP",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "PDS"
            }
        }
    ]
}
```

Where:

- **meta.project.name**: representing the name of AD project.
- **meta.project.uuid**: representing the UUID of the AD project.
- **meta.snapshot**: representing the UNIX timestamp (in seconds) at the time the data was collected.
- **data**: representing an array containing all the CICS regions artifacts.

The dataset artifact:

- **uid**: representing the string unique identifier.
- **name**: representing the dataset name.
- **type**: representing the DATASET (the artifact type).
- **properties.volumeSerial**: representing the **Volume** parameter from the DD definition.
- **properties.datasetType**: representing the dataset type (**PDS**, **GDG** or **DATASET**)

GET /projects/{project-uuid}/cics

Where *{project-uuid}* is a unique identifier for a project.

Description: Get all CICS regions available in an AD project.

Sample request:

```
curl --user username:password https://{host}:{port}/projects/SAMPLE-PROJECT/cics
```

Sample response:

```
{
    "meta": {
        "name": "SAMPLE-PROJECT",
        "uuid": "SAMPLE-PROJECT",
        "snapshot": 1513610223
    },
    "data": [
        {
            "uid": "56fa655b88f42e49246da037a2dabee69027befa07b31e1774464eb3c93bba0e",
            "name": "CICSTS53",
            "type": "REGION"
        },
        {
            "uid": "90e806231100cb8108d0b9a639cc5fd2cd92689c4b40d051329534a881fc6ea1",
            "name": "CICSTS52",
            "type": "REGION"
        }
    ]
}
```

Where:

- **meta.project.name**: representing the name of AD project.
- **meta.project.uuid**: representing the UUID of the AD project.
- **meta.snapshot**: representing the UNIX timestamp (in seconds) when the data was collected.
- **data**: representing an array containing all the CICS regions artifacts.

The CICS region artifact:

- **uid**: representing a string unique identifier
- **name**: representing the region name.
- **type**: representing the *REGION* (the artifact type).

GET /projects/{project-uuid}/cics/{region-uid}/datasets

Where:

- **{project-uuid}**: is a unique identifier for a project.
- **{region-uid}**: is a unique identifier of the a cics region artifact.

Description: Get all datasets used by a cics application.

Sample request:

```
curl --user username:password https://{host}:
{port}/projects/90e806231100cb8108d0b9a639cc5fd2cd92689c4b40d051329534a881fc6ea1/datasets
```

```
{
    "meta": {
        "project": {
            "name": "SAMPLE-PROJECT",
            "uuid": "SAMPLE-PROJECT",
        },
        "snapshot": 1502354654
    },
    "data": [
        {
            "uid": "01da56d83ec1c04be901ebe162957e3d3eaf6e49a85a0f31497f535b1f567b32",
            "name": "CB12V51.GENAPP.KSDSCUST.REPORT",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "DATASET"
            }
        },
        {
            "uid": "09c2ef33f622e62da876e1d73a7e9e8f3aa8236034973a3f232acbf4da8509ee",
            "name": "DFH520.CICS.SDFJAUTH",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "DATASET"
```

```
            }
        },
        {
            "uid": "0c296329019f68889434bc14103de4475c3a680c64ee91f9a8f331c736b30257",
            "name": "DFH530.CNTL.CICS.DFHTEMP",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "DATASET"
            }
        }
    ]
}
```

Where:

- **meta.project.name**: representing the name of AD project.
- **meta.project.uuid**: representing the UUID of the AD project.
- **meta.snapshot**: representing the UNIX timestamp (in seconds).

GET /projects/{project-uuid}/jobs

Description: Get all jobs available in an AD project.

Sample request:

```
curl --user username:password https://{host}:{port}/projects/SAMPLE-PROJECT/jobs
```

Sample response:

```
{
    "meta": {
        "project": {
            "name": "SAMPLE-PROJECT",
            "uuid": "SAMPLE-PROJECT",
        },
        "snapshot": 1502354654
    },
    "data": [
        {
            "uid": "0b934ea4cfb7f446c7d926e711c45e12383bbfbea8d030dd34a154c7e3974076",
            "name": "CICSTS53",
            "type": "JCL"
        },
        {
            "uid": "5763b2fdc5a06f4af67e65f994585bdbe2eb00a7fc9aa72c58195d4906eda17f",
            "name": "CRTAIXS",
            "type": "JCL"
        },
        {
            "uid": "6b6f6023b61db6ee87798a6ecb6272aec0cf44b10d609e9ffe3807c7378dd786",
            "name": "RUNCOBOL",
            "type": "JCL"
        }
    ]
}
```

Where:

- **meta.project.name**: representing the name of AD project.
- **meta.project.uuid**: representing the UUID of the AD project.
- **meta.snapshot**: representing the UNIX timestamp (in seconds) when the data was collected.
- **data**: representing an array containing all the job artifacts.

The job artifact:

- **uid**: representing a string unique identifier.
- **name**: representing the job name.
- **type**: representing the JCL (the artifact type) at the time the data was collected.

- **data**: representing an array containing all the datasets artifacts in the CICS region identified by **{region-uid}**.

GET /projects/{project-uuid}/jobs/{job-uid}/datasets

Description: Get all datasets defined in DD cards for a given Job.

Sample request:

```
curl --user username:password https://{host}:{port}/projects/SAMPLE-
PROJECT/jobs/0b934ea4cfb7f446c7d926e711c45e12383bbfbea8d030dd34a154c7e3974076/datasets
```

Sample response:

```
{
    "meta": {
        "project": {
            "name": "SAMPLE-PROJECT",
            "uuid": "SAMPLE-PROJECT",
        },
        "snapshot": 1502354654
    },
    "data": [
        {
            "uid": "01da56d83ec1c04be901ebe162957e3d3eaf6e49a85a0f31497f535b1f567b32",
            "name": "CB12V51.GENAPP.KSDSCUST.REPORT",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "(,,,4)",
                "datasetType": "GDG"
            }
        },
        {
            "uid": "09c2ef33f622e62da876e1d73a7e9e8f3aa8236034973a3f232acbf4da8509ee",
            "name": "DFH520.CICS.SDFJAUTH",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "DATASET"
            }
        },
        {
            "uid": "0c296329019f68889434bc14103de4475c3a680c64ee91f9a8f331c736b30257",
            "name": "DFH530.CNTL.CICS.DFHTEMP",
            "type": "DATASET",
            "properties": {
                "volumeSerial": "",
                "datasetType": "PDS"
            }
        }
    ]
}
```

Where:

- **meta.project.name**: representing the name of AD project.
- **meta.project.uuid**: representing the UUID of the AD project.
- **meta.snapshot**: representing the UNIX timestamp (in seconds) when the data was collected.
- **data**: representing an array containing all the datasets artifacts defined in DD cards for the job identified by **{job-uid}**.

Sample JCL and PROCS for IBM AD Connect for Mainframe

Within the sample library (IBM.HALT505.F3.BIN), after it is uncompressed, the following individual members are provided:

- **IAYLSTNR** – a sample PROC for the IBM AD Connect for Mainframe listener started task. This PROC should be configured depending on the number of started tasks required and the SCM product that has to be accessed (eg. Endevor, ChangeMan ZMF, and so on). See the sections that follow for detailed explanations about how to configure the listener proc.

- **IAYDB2BD** – a sample job for creating the DBRM plan, required only if IBM AD Connect for Mainframe should access Db2.
- **IAYCA7** – a sample JOB for accessing CA7 (if not accessing in memory, but in batch)
- **IAYNATDB** – a sample JOB for accessing Adabas.
- **IAYNATLB** – a sample JOB for accessing Natural.
- **IAYNATMM** – a sample JOB for accessing Natural.
- **IAYXMLRQ** – a sample JOB for accessing ChangeMan ZMF.

Chapter 12. Troubleshooting

Known issues and workarounds

Find instructions and solutions to the known issues that you might encounter with IBM Application Discovery.

1. All IBM AD services are stopped and not restarted as expected when you reboot the Linux machine

When you reboot the Windows machine, all the services would be restarted automatically. However, when you reboot the Linux machine, you find that none of the AD services get back up and running, which is a limitation to the Linux support at the moment. The workaround is to manually restart the four main AD services in order when the Linux machine completes its reboot.

Check the following list for the four main AD services and the commands to manually start each service.

- **IBM Service Manager**: `<AD_INSTALL_DIRECTORY>/IBM Application Discovery Service Manager/startServer.sh`
- **IBM AD Web Services**:`<AD_INSTALL_DIRECTORY>/IBM AD Web Services/wlp/bin/server start ad_server`
- **IBM Configuration Service**: `<AD_INSTALL_DIRECTORY>/IBM Application Discovery Configuration Service/startServer.sh`
- **IBM Configuration Admin Service**:`<AD_INSTALL_DIRECTORY>/IBM Application Discovery Configuration Service/startWebService.sh`

After all the four main AD services are started successfully, you can start to use AD again.

2. IBM ADDI Service Manager does not automatically start after the configuration or upgrade process of IBM AD on Linux Red Hat 8 machine

After the configuration or upgrade process, it might be possible that ADDI Service Manager does not start automatically, which would cause **IBM AD Configuration Wizard** and **IBM AD Upgrade Wizard** not to complete successfully. The workaround is to manually start IBM ADDI Service Manager.

To start **IBM ADDI Service Manager**, proceed as follows:

1. Go to `<IBM ADDI Installation Folder>/IBM Application Discovery Service Manager/` and locate the `startServer.sh` file.
2. In case that the `.sh` file is not executable, open a terminal and run the following command for flagging them as executable.

   ```
   chmod +x startServer.sh
   ```

3. If the service does not start, check the `ad-service-manager.log` file located in the `<IBM ADDI Installation Folder>/IBM Application Discovery Service Manager` folder.

3. When upgrading from IBM AD V5.1.0.9 to IBM AD V5.1.0.9 iFix 1 on Linux Red Hat 8 machine, it might be required to set a new password for IBM AD GraphDB

In case that you have upgraded IBM AD from IBM AD V5.1.0.9 to IBM AD V5.1.0.9 iFix 1 on Linux Red Hat® 8 machine and you have decided to change the default IBM AD GraphDB port, it is possible to cause IBM AD GraphDB to not work correctly. The workaround is to manually set a new password and confirm it.

To set a new password for **IBM AD GraphDB**, proceed as follows:

© Copyright IBM Corp. 2010, 2022

1. Access **Start Menu** > **IBM Application Discovery and Delivery Intelligence** > **Launch IBM Application Discovery Configuration Service Admin**, and go to **Configure** > **Environments**.
2. Set a new password and confirm it.
3. Click **Save**.

Messages

This reference information provides additional information about messages you might encounter while using the product. It is organized according to the identifier of the product feature that produces the message.

The message identifier can be 10 characters in length and has the form:

```
CCCCC1234X
```

where:

CCCCC
is a 5-character alphabetic component identifier.

1234
is a 4-character numeric identifier used to identify the specific message for that component.

X
is an optional alphabetic severity indicator. (I=Informational, W=Warning, E=Error)

IBM AD Connect for Mainframe Messages

IAYMF0001E **IBM AD Connect for Mainframe Connector is not APF authorized.**

Explanation:
All the load libraries in 'STEPLIB' of the IBM AD Connect for Mainframe started task procedure must be authorized by authorized program facility (APF).

System action:
IBM AD Connect for Mainframe started task terminates.

User response

Ensure that all libraries in the DDNAME STEPLIB are APF authorized. To add a library to APF, use this operator command for a library:

```
SETPROG APF,ADD,DSNAME=libname,VOLUME=volume
```

Consult with your system programmer to ensure that the change is made permanent.

IAYMF0005I **IBM AD Connect for Mainframe Connector is active on port {0}.**

Explanation

This message displays the 5-digit TCP/IP port number for the IBM AD Connect for Mainframe started task. This port number is used in IBM ADDI configuration for sending requests to IBM AD Connect for Mainframe to process. Example:

```
IAYMF0005I  IBM AD Connect for Mainframe
Connector is active on port 05105.
```

System action:
IBM AD Connect for Mainframe listens for incoming requests on the specified port.

User response:
When defining a z/OS connection, ensure that IBM AD Build Client is configured to use this port. For more information, see “Configuring the z/OS Connection” on page 216.

IAYMF0006E **IBM AD Connect for Mainframe Connector is terminating. TCP/IP is not active.**

Explanation:
The TCP/IP address space is not active (running). IBM AD Connect for Mainframe cannot function without an

active TCP/IP stack on the z/OS system on which it runs.

System action:
IBM AD Connect for Mainframe started task terminates.

User response

Verify that the TCP/IP address space is up and running, and that the address space name is specified in the IBM AD Connect for Mainframe parameters correctly. To get the TCP/IP address space name, use the **NETSTAT HOME** command. Example:

```
TSO NETSTAT HOME

MVS TCP/IP NETSTAT CS V1R13       TCPIP Name:
TCPIP           19:36:27
User Id  Conn     State
```

After the address name is verified, it can be set in the AD Connect for Mainframe started task procedure JCL, in the **TCPIP** keyword parameter. It is common that the name of the address space is also TCPIP, as it is in the following example:

```
//IAYLSTNR PROC
TCPIP=TCPIP,PORT=46000,MAXTASK=20,PROGRAM=IAYCON
N,
//          HOSTCP=00037,CLNTCP=00850
```

IAYMF0007I IBM AD Connect for Mainframe Connector parameters selected:

Explanation:
The IBM AD connect for Mainframe Connector parameters are displayed.

System action:
IBM AD Connect for Mainframe started task continues the startup sequence.

User response:
No action is required.

IAYMF0008I TCPIP address space name: {0}.

Explanation

This message displays the TCP/IP address space name. Example:

```
IAYMF0008I       TCPIP address space name:
TCPIP   .
```

System action:
IBM AD Connect for Mainframe started task continues the startup sequence.

User response:
No action is required.

IAYMF0009I port number: {0}.

Explanation

This message displays the 5-digit port number. Example:

```
IAYMF0009I       port number: 05105.
```

System action:
IBM AD Connect for Mainframe started task continues the startup sequence.

User response:
No action is required.

IAYMF0010I maximum tasks: {0}.

Explanation

This message indicates the maximum number of subtasks that can be started by IBM AD Connect for Mainframe to process client requests. Example:

```
IAYMF0010I       maximum tasks: 0005.
```

System action:
IBM AD Connect for Mainframe started task continues the startup sequence.

User response:
No action is required.

IAYMF0011I program name: {0}.

Explanation

This message displays the program name. Example:

```
IAYMF0011I       program name: IAYCONN .
```

System action:
IBM AD Connect for Mainframe started task continues the startup sequence.

User response:
No action is required.

IAYMF0012E IBM AD Connect for Mainframe Connector error: parameters missing.

Explanation:
Some IBM AD Connect for Mainframe mandatory parameters are missing on the started task procedure JCL.

System action:
IBM AD Connect for Mainframe started task terminates.

User response

A sample of the mandatory parameters is provided in member IAYLSTNR of library SIAYSAMP. To ensure that all the parameters are set in the IBM AD Connect for Mainframe started task procedure JCL, customize

the IAYLSTNR member, and then save it in one of your system's procedure libraries. The following example shows a complete set of parameters, which are positional parameters separated by commas:

```
PARM='&TCPAS,&PORT,&MAXTK,&CONNNAME,&DEB,&HOSTCP
,&CLNTCP,&DBC,&RXI'
```

&TCPAS
Specifies the TCP/IP address space name, for example, TCPIP.

&PORT
Specifies the TCP/IP port number, for example, 05105. This value must be a five-digit number, with leading zeros if required.

&MAXTK
Specifies the maximum number of tasks. The value must be a number in the range 1 - 20.

&CONNNAME
Specifies the program name. The value must be set to IAYCONN.

&DEB
Indicates the debug mode. The following debug modes are available:

Value	Debug mode	Explanation
Y	Debug on UTC	The long form with coordinated universal time (UTC) timestamps. Long form debug output contains detailed tracing of all the operations that are performed by IBM AD Connect for Mainframe. Because this mode generates large volumes of output, use this mode only for a short duration.
N	Disable debugging	All debug output is stopped.
L	Debug on local time	The long form with local time (LT) timestamps. Long form debug output contains detailed tracing of all the operations that are performed by IBM AD Connect for Mainframe. Because this mode generates large volumes of output, use this mode only for a short duration.
S	Debug on short UTC	The short form with UTC timestamps. To minimize the volume of debug output that is written to the system spool, short form debug output displays only the essential steps that are taken by the code.
T	Debug on short local time	The short form with LT timestamps. To minimize the volume of debug output that is written to the system spool, short form debug output displays only the essential steps that are taken by the code.

&HOSTCP
Specifies the CCSID of the host code page, for example, 00278. This value must be a five-digit number, with leading zeros if required.

&CLNTCP
Specifies the CCSID of the client code page, for example, 01252. This value must be a five-digit number, with leading zeros if required.

&DBC
This parameter is obsolete, but must remain in this position to preserve the correct position of the parameters that follow. The value must be set to N, which is used as a placeholder.

&RXI
Indicates whether to run CICS CSD retrieve in the batch mode. Set the value to Y for yes, or N for no.

IAYMF0013E IBM AD Connect for Mainframe Connector error: port number length more than 5.

Explanation:
The port number contains more than five digits.

System action:
IBM AD Connect for Mainframe started task terminates.

User response
Set the port number on the second positional parameter in the IBM AD Connect for Mainframe started task JCL to a 5-digit value. Example:

```
PARM='TCPIP,05105,5,IAYCONN,T,00278,01252,N,Y'
```

IAYMF0014E IBM AD Connect for Mainframe Connector error: tcpip name length more than 8.

Explanation:
The TCP/IP address space name contains more than eight characters.

System action:
IBM AD Connect for Mainframe started task terminates.

User response
Set the TCP/IP address space name on the first positional parameter in the IBM AD Connect for Mainframe started task JCL to a value that contains no more than eight characters. Example:

```
PARM='TCPIP,05105,5,IAYCONN,T,00278,01252,N,Y'
```

To get the TCP/IP address space name, use the **NETSTAT HOME** command.

IAYMF0015E IBM AD Connect for Mainframe Connector error: max. task number length more than 4.

Explanation:
The maximum number of tasks contains more than four digits. The acceptable range is 1 - 20.

System action:
IBM AD Connect for Mainframe started task terminates.

User response
Set the maximum number of tasks on the third positional parameter in the IBM AD Connect for Mainframe started task JCL to a value that is no bigger than 20. Example:

```
PARM='TCPIP,05105,5,IAYCONN,T,00278,01252,N,Y'
```

IAYMF0016E IBM AD Connect for Mainframe Connector error: progam name length greater than 8.

Explanation:
The program name contains more than eight characters. The default value is IAYCONN, and this is the only valid value.

System action:
IBM AD Connect for Mainframe started task terminates.

User response
Set the program name on the fourth positional parameter in the IBM AD Connect for Mainframe started task JCL to IAYCONN. Example:

```
PARM='TCPIP,05105,5,IAYCONN,T,00278,01252,N,Y'
```

IAYMF0017E IBM AD Connect for Mainframe Connector: unrecognized operator modifier.

Explanation:
The command input buffer (CIB) command code cannot be recognized.

System action:
The incorrect command is echoed in the IBM AD Connect for Mainframe started task message log.

User response:
Correct the command.

IAYMF0018I IBM AD Connect for Mainframe Connector terminated upon operator request.

Explanation:
IBM AD Connect for Mainframe is being terminated normally according to operator request.

System action:
IBM AD Connect for Mainframe terminates.

User response:
No action is required.

IAYMF0019I **IBM AD Connect for Mainframe Connector shutting down due to operator request.**

Explanation:
IBM AD Connect for Mainframe is being terminated normally according to operator request.

System action:
IBM AD Connect for Mainframe terminates.

User response:
No action is required.

IAYMF0020I **IBM AD Connect for Mainframe Connector shutdown complete.**

Explanation:
IBM AD Connect for Mainframe is terminated normally.

System action:
IBM AD Connect for Mainframe terminates.

User response:
No action is required.

IAYMF0021W **IBM AD Connect for Mainframe Connector invalid modify option: {0}.**

Explanation:
The modify command suboption is invalid.

System action:
The command is rejected.

User response

Use a valid modify command. Example:

```
/F IAYV6109,DEBUGON
```

IAYV6109
The name of the IBM AD Connect for Mainframe started task.

DEBUGON
Sets the debug mode on.

IAYMF0023E **Error code returned from TCP. func: {0} error code is: {1}.**

Explanation:
The error that is returned from TCP for the requested function {0} and error code {1}.

System action:
Depending on the severity or nature of the error, IBM AD Connect for Mainframe will attempt to recover.

User response:
For explanations and possible solutions, refer to the z/OS TCP/IP error code, which is documented in *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*. If the error persists and cannot be corrected, contact IBM Support.

IAYMF0024E **IBM AD Connect for Mainframe Connector terminated with error:**

Explanation:
The error that is returned from TCP with the function name and error code.

System action:
IBM AD Connect for Mainframe terminates with error.

User response:
For explanations and possible solutions, refer to the z/OS TCP/IP error code, which is documented in *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*. If the error persists and cannot be corrected, contact IBM Support.

IAYMF0025I **Specified port is in use by another address space or user.**

Explanation:
The IBM AD Connect for Mainframe task cannot start because the specified port is in use by another TCP/IP address space or user.

System action:
IBM AD Connect for Mainframe terminates with error.

User response:
Ask your network team to assign and reserve a port that can be used exclusively by the IBM AD Connect for Mainframe started task. To list all the ports that are currently in use on the system, use the **NETSTAT** command.

IAYMF0026I **Check that the name of the TCP/IP address space is correct.**

Explanation:
The IBM AD Connect for Mainframe task cannot start because the specified TCP/IP address space is not found.

System action:
IBM AD Connect for Mainframe terminates with error.

User response

Verify that the TCP/IP address space is up and running, and that the address space name is specified in the IBM AD Connect for Mainframe parameters correctly. To get the TCP/IP address space name, use the **NETSTAT HOME** command. Example:

```
TSO NETSTAT HOME

MVS TCP/IP NETSTAT CS V1R13       TCPIP Name:
```

```
TCPIP           19:36:27
User Id  Conn     State
```

After the address name is verified, it can be set in the IBM AD Connect for Mainframe started task procedure JCL, in the **TCPIP** keyword parameter. It is common that the name of the address space is also TCPIP, as it is in the following example:

```
//IAYLSTNR PROC
TCPIP=TCPIP,PORT=46000,MAXTASK=20,PROGRAM=IAYCON
N,
//          HOSTCP=00037,CLNTCP=00850
```

IAYMF0027I Security Error: Add an OMVS segment to the user assigned to the Connector.

Explanation:
The service ID that is used for the IBM AD Connect for Mainframe started task does not have an OMVS segment. An OMVS segment must be added to the service ID for the started task in RACF or an equivalent security product.

System action:
IBM AD Connect for Mainframe terminates with error.

User response:
Request your security administrator to add an OMVS segment to the IBM AD Connect for Mainframe started task user ID. For more information, see topic "Steps for defining z/OS UNIX users to RACF" in the IBM z/OS documentation.

IAYMF0028I Check if a different TCP/IP application is using the same jobname.

Explanation:
A started task or batch job with the same name as the IBM AD Connect for Mainframe task is already in session with the TCP/IP address space.

System action:
IBM AD Connect for Mainframe terminates with error.

User response:
Investigate why there is another job or task instance with the same name as IBM AD Connect for Mainframe. If the other job or task is legitimate, choose another job name for IBM AD Connect for Mainframe.

IAYMF0029I Check that TCP/IP is active.

Explanation:
The TCP/IP address space that is specified in the IBM AD Connect for Mainframe parameters is not active.

System action:
IBM AD Connect for Mainframe terminates with error.

User response:
Request your system programmer to confirm that TCP/IP is active on the z/OS system upon which IBM AD Connect for Mainframe is attempting to start. Confirm the name of the TCP/IP address space.

IAYMF0030I Check that the port is not reserved to another task.

Explanation:
The IBM AD Connect for Mainframe task cannot start because the specified port is in use by another TCP/IP address space or user.

System action:
IBM AD Connect for Mainframe terminates with error.

User response:
Ask your network team to assign and reserve a port that can be used exclusively by the IBM AD Connect for Mainframe started task. To list all the ports that are currently in use on the system, use the **NETSTAT** command.

IAYMF0032I DEBUG option for IBM AD Connect for Mainframe CONNECTOR is SHORT (UTC)

Explanation:
This message is to confirm that the debug mode is set to the short form with coordinated universal time (UTC) timestamps. To minimize the volume of debug output that is written to the system spool, short form debug output displays only the essential steps that are taken by the code.

System action:
Short form debug output with UTC timestamps is generated to DDNAME IAYOUT.

User response:
No action is required.

IAYMF0033I IBM AD Connect for Mainframe version: {0} compile date {1}.

Explanation:
This message displays the version number of IBM AD Connect for Mainframe and the compile date.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0034I DEBUG option for IBM AD Connect for Mainframe CONNECTOR is ON (UTC)

Explanation:
This message is to confirm that the debug mode is set to the long form with coordinated universal time (UTC) timestamps. Long form debug output contains detailed tracing of all the operations that are performed by IBM AD Connect for Mainframe. Because this mode

generates large volumes of output, use this mode only for a short duration.

System action:
Long form debug output with UTC timestamps is generated to DDNAME IAYOUT.

User response:
No action is required.

IAYMF0035I DEBUG option for IBM AD Connect for Mainframe CONNECTOR is OFF

Explanation:
This message is to confirm that the debug mode is set to OFF, which means all debug output is stopped.

System action:
All debug output is stopped and no longer written to DDNAME IAYOUT.

User response:
No action is required.

IAYMF0036I OPERATOR COMMAND SUBTASK IS BEING STARTED

Explanation:
This message confirms that the subtask for handling operator commands is starting.

System action:
The operator command subtask is starting.

User response:
No action is required.

IAYMF0037I OPERATOR COMMAND SUBTASK STARTED

Explanation:
This message confirms that the subtask for handling operator commands is started.

System action:
Operator command subtask is started.

User response:
No action is required.

IAYMF0038E Error code returned from TCP. func: {0} return code is: {1}. errcode is: {2}

Explanation:
The error that is returned from TCP with the function name and error code.

System action:
Depending on the severity or nature of the error, IBM AD Connect for Mainframe will attempt to recover.

User response:
For explanations and possible solutions, refer to the z/OS TCP/IP error code, which is documented in *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*. If the error persists and cannot be corrected, contact IBM Support.

IAYMF0039I TRANSLATE TABLES FOR JAPANESE IS INITIATED

Explanation:
This message confirms that the double-byte character set translation tables for the Japanese language are enabled.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0040I HOST PAGE CODE IS: {0}

Explanation:
This message indicates the host code page that is used when translating text from the mainframe code page to the client code page.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0041I CLIENT PAGE CODE IS: {0}

Explanation:
This message indicates the client code page that is used when translating text from the mainframe code page to the client code page.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0042I MODULE IAYCONN HAS BEEN RELOADED SUCCESSFULLY AT {0}

Explanation:
This message is displayed when the system must reload the IAYCONN module at a different location in memory. It occurs at the operating system level. This is not something that IBM AD Connect for Mainframe can control.

System action:
IAYCONN is called at a new address.

User response:
No action is required.

IAYMF0043I DEBUG option for IBM AD Connect for Mainframe CONNECTOR is ON (LT)

Explanation:
This message is to confirm that the debug mode is set to the long form with local time (LT) timestamps. Long form debug output contains detailed tracing of all

the operations that are performed by IBM AD Connect for Mainframe. Because this mode generates large volumes of output, use this mode only for a short duration.

System action:
Long form debug output with LT timestamps is generated to DDNAME IAYOUT.

User response:
No action is required.

IAYMF0044I DEBUG option for IBM AD Connect for Mainframe CONNECTOR is SHORT (LT)

Explanation:
This message is to confirm that the debug mode is set to the short form with local time (LT) timestamps. To minimize the volume of debug output that is written to the system spool, short form debug output displays only the essential steps that are taken by the code.

System action:
Short form debug output with LT timestamps is generated to DDNAME IAYOUT.

User response:
No action is required.

IAYMF0045I {0}, Invalid length in RECV - x'{1}'.

Explanation:
An unexpected result that is received during the TCP/IP receive operation. This might be a temporary or one time error occurrence. This might be caused by a network problem, or the IBM AD Build Client configuration dialog or IBM AD Build Client itself terminating unexpectedly.

System action:
The IBM AD Connect for Mainframe started task will recover.

User response:
If the messages persist and there is an observed problem with the IBM AD Build Client communication to the IBM AD Connect for Mainframe started task, contact IBM Support. Provide the started task job output and any error messages that are observed from the IBM AD Build Client side.

IAYMF0046I REXX INTEGRATION IS ON.

Explanation:
This message is to confirm that REXX integration is turned on. REXX integration is a feature that allows the IBM AD Connect for Mainframe started task to more flexibly handle batch interfaces to products like CA7.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0047I REXX INTEGRATION IS OFF.

Explanation:
This message is to confirm that REXX integration is turned off. REXX integration is a feature that allows the IBM AD Connect for Mainframe started task to more flexibly handle batch interfaces to products like CA7.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0048I -- {0} lines read from IAYRPFIL on {1} at {2}. No error.

Explanation:
This message is to confirm that the optional configuration file specified by `DDNAME IAYRFFIL` is located, and to indicate the number of lines that are read from the configuration file. The settings in this file will override any settings specified in the PROC parameters.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF0050I --Dynamic parameters refresh disabled - no IAYRFFIL DD.

Explanation:
This message is to confirm that no optional configuration file is specified by `DDNAME IAYRFFIL`.

System action:
IBM AD Connect for Mainframe continues.

User response:
No action is required.

IAYMF1003E DATA SET OPEN ERROR, RC=*{0}* RS=*{1}* DATA SET=*{2}*

Explanation

IBM AD Connect for Mainframe failed to open the indicated data set for READ access. Where:

- *{0}* - represents the **OPEN return code** in hexadecimal.
- *{1}* - represents the **OPEN reason code** in hexadecimal.
- *{2}* - represents the name of the data set for which the OPEN call failed.

System action:
The user request function is not completed.

User response

Confirm that the data set that is being opened is the correct one based on the type of request sent to IBM AD Connect for Mainframe and the configuration of IBM AD Build Client.

Description of all possible OPEN return and reason codes can be found at https://www.ibm.com/docs/en/zos/2.4.0?topic=codes-open-return-reason.

If the error persists and cannot be corrected, contact IBM Support.

IAYMF1004E DATA SET ALLOCATION ERROR, RC={0} RS={1} DATA SET={2}

Explanation

IBM AD Connect for Mainframe failed to dynamically allocate the indicated data set. IBM AD Connect for Mainframe must allocate a data set before it can open it for input (READ only). Where:

- *{0}* - represents the **DYNALLOC return code** in decimal.
- *{1}* - represents the **DYNALLOC reason code** in hexadecimal.
- *{2}* - represents the name of the data set for which dynamic allocation failed.

System action:
The user request function is not completed.

User response

Confirm that the data set that is being dynamically allocated is the correct one based on the type of request sent to IBM AD Connect for Mainframe and the configuration of IBM AD Build Client.

Description of DYNALLOC return and reason codes can be found at:

- https://www.ibm.com/docs/en/zos/2.4.0?topic=functions-interpreting-dynalloc-return-codes
- https://www.ibm.com/docs/en/zos/2.4.0?topic=codes-interpreting-information-reason-from-dynalloc
- https://www.ibm.com/docs/en/zos/2.4.0?topic=codes-interpreting-error-reason-from-dynalloc

If the error persists and cannot be corrected, contact IBM Support.

IAYMF1041E {0}

Explanation:
The error message that is returned from Db2 during connection attempt. The most common reason for this error is that the Db2 bind was not successful when IBM AD Connect for Mainframe was installed, or the correct permission was not granted to the IBM AD Connect for Mainframe started task user ID.

System action:
Connection fails with error. IBM AD Connect for Mainframe returns the Db2 error message to IBM AD Build Configuration. The message is also saved in the MVS error logs folder that is associated with the z/OS connection and the current IBM AD project.

User response:
Confirm that the IBM AD Connect for Mainframe started task user ID is granted the EXECUTE authority on the IAYSRC plan. If the reason for the connection failure is not clear from the message, check the Db2 documentation, or contact IBM Support.

IAYMF1084E USERID AND/OR PASSWORD NOT PROVIDED.

Explanation:
This error message indicates that IBM AD Build Client is not configured to send the user ID and password but IBM AD Connect for Mainframe is configured to expect the credentials.

System action:
IBM AD Connect for Mainframe does not process the request.

User response:
Correct the discrepancy. Configure both client and server to enable or disable **z/OS User Authentication**.

IAYMF1085E AUTHENTICATION FAILED FOR USERID *XXXXXXX*

Explanation:
This error message indicates that either the entered user ID or the password is not correct.

System action:
IBM AD Connect for Mainframe does not process the request.

User response

Correct the user ID or the password that is entered.

More detailed informational messages are also displayed in the IBM AD Connect for Mainframe **JESMSGLG** and system log. For example, the messages will indicate whether the user ID or the password is incorrect.

```
ICH408I USER(USERX   ) GROUP(        )
NAME(???                 )
  LOGON/JOB INITIATION - USER AT
TERMINAL          NOT RACF-DEFINED
IRR012I  VERIFICATION FAILED. USER PROFILE NOT
FOUND.

ICH408I USER(USER2   ) GROUP(SYS1    )
NAME(######################)
  LOGON/JOB INITIATION - INVALID
PASSWORD
```

```
IRR013I  VERIFICATION FAILED. INVALID PASSWORD
GIVEN.
```

IAYMF1086E **AD Connect for Mainframe not configured to accept userid/password sent from AD Build Client**

Explanation:
This error message indicates that IBM AD Connect for Mainframe is not configured to accept user ID and password, while IBM AD Build Client is configured to send them.

System action:
IBM AD Connect for Mainframe does not process the request.

User response:
Correct the discrepancy. Configure both client and server to enable or disable **z/OS User Authentication**.

IBM AD Build Client Messages

Finding Log Files

The following table summarizes a list of log files that are generated by a specific component. For monitoring any component's tasks, check the logs that are located in their specific location.

Table 43. Log files Location

Component	Logs location	Log files name format
IBM AD Configuration Server	<IBM ADDI Installation Folder>\IBM Application Discovery Configuration Service\log\	• server.log • ibmapplicationdiscoveryconfigurationadminservice-stderr.<date>.log • ibmapplicationdiscoveryconfigurationadminservice-stdout.<date>.log • ibmapplicationdiscoveryconfigurationservice-stderr.<date>.log • ibmapplicationdiscoveryconfigurationservice-stdout.<date>.log • ezservice-daemon.<date>.log • ws-service-daemon.<date>.log

Table 43. Log files Location (continued)

Component	Logs location	Log files name format
IBM AD Build Client	<IBM AD Build Client installation folder>\Bin\Release\	**IBM AD Build Client** can be invoked in batch mode to create a new project in background. The log for the **New project in background** has the following name format NewProjInBackLog_<timestamp>.log.
	<Project's Folder>	**IBM AD Build Client** can be invoked in batch mode to run a build (full build). The log for the **Build in background** process has the following name format Project'sName_<timestamp>.txt.
		IBM AD Build Client can be invoked in batch mode to keep the information stored in repository up-to-date. The logs for the **Make** process have the following format: • BatchMakeStatusFile_<timestamp>.txt • Project'sName_<timestamp>.txt
		IBM AD Build Client can automatically add all files from a given physical folder to a virtual one (similar to Add all files from folder option that can be found in GUI Mode). The log for this process has the following format UpdateInBackgroundLog_<timestamp>.txt.
		IBM AD Build Client can be invoked in batch mode to update **API Resolutions**. The log for this process has the following format UpdateApiResolutions_<timestamp>.txt.
	<Project's Folder>\reports\	BuildErrorReport_<timestamp>.html is generated by any build performed in **IBM AD Build Client**.
	<Project's Folder>\Synchronize\	• ConfigFileValidation_<timestamp>.log • SynchronizeMembersExtendedInfo_<timestamp> • SynchronizeMembersProgress_<timestamp> • SynchronizeMembersSummary_<timestamp>
	<Project's Folder>\MVS\Logs\<zOS Connection's Name>	• MFImport_<timestamp>.log • MFErrors_<timestamp>.log
	<IBM ADDI Installation Folder>\IBM Application Discovery Build Client\Bin\Release\CCS_Interaction_Logs\	The log for the interaction with **IBM AD Configuration Server** can be found under CCSErrors_<timestamp>.log.
	<Path For The Retrieved Members>\Mainframe Library Members\	• BackgroundUpdateMFMemberSources_<timestamp>.log • SummaryUpdateMFMemberSources_<timestamp>.log • SummaryGetMFMemberSources_<timestamp>.log
	<Path For The Retrieved Members>\Mainframe Library Members\<zOS Connection's Name>	• GetMFMemberSources_<timestamp>.log • UpdateMFMemberSources_<timestamp>.log
IBM AD Build Configuration	<IBM AD Build Client installation folder>\Bin\Release\Log\	ADBuildConfiguration_<timestamp>.log
IBM Application Discovery GraphDB	<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\orientdb\log\	ezservice-daemon.<date>.log
	<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\orientdb-community-*x.x.x*_ezpatch1\log\	orient-server.log

Table 43. Log files Location (continued)

Component	Logs location	Log files name format
IBM AD Validation Server	<IBM AD Build Client installation path>\bin\Release\IBMApplication DiscoveryValidationServer\Logs\	• ValidationServiceActivity_<number>.log • ValidationServiceStatus_<timestamp>.log
	<Project's Folder>\Validation\	ValidationThroughPDS_GenerateIncludePaths_<timestamp>.log
	<Project's Folder>\Synchronize\	
Authentication Server (DEX)	<IBM ADDI Installation Folder>\Authentication Server (DEX)\	dex.log
IBM AD File Service	<IBM ADDI Installation Folder>\IBM Application Discovery File Service\	.log
IBM AD Manual Resolution Service	<IBM ADDI Installation Folder>\IBM Application Discovery Manual Resolutions Service\log\	manualres.log
IBM AD Mainframe Projects	<IBM ADDI Installation Folder>\IBM Application Discovery Mainframe Projects Service\log\	mfprojs.log
IBM AD Batch Server	<IBM ADDI Installation Folder>\IBM Application Discovery Batch Server\log\	• server.log • ibmapplicationdiscoverybatchservice -stdout.<date>.log • ibmapplicationdiscoverybatchservice -stderr.<date>.log • ibmapplicationdiscoverywebservice -stdout.<date>.log • ibmapplicationdiscoverywebservice -stderr.<date>.log • server-daemon.<date>.log • webservice.log • webservice-daemon.<date>.log • status.log • ProjectName-projectDBVersion.log • gdbTool.log • rb.log
IBM AD Search Service	<IBM ADDI Installation Folder>\IBM Application Discovery Search Service\log\	search.log
IBM AD Cross Applications Service	<IBM ADDI Installation Folder>\IBM Application Discovery Cross Applications Service\log\	cross.log

Table 43. Log files Location (continued)		
Component	**Logs location**	**Log files name format**
IBM AD Analyze Server	<IBM ADDI Installation Folder>\IBM Application Discovery Analyze Server\log\log_service\	• ezservice-daemon.<date>.log • ibmapplicationdiscoveryanalyzeservice-stderr.<date>.log • ibmapplicationdiscoveryanalyzeservice-stdout.<date>.log
	<IBM ADDI Installation Folder>\IBM Application Discovery Analyze Server\log\server\	• server.log • server.<date>.log
	<IBM ADDI Installation Folder>\IBM Application Discovery Analyze Server\log\manager\	• manager.log • manager.<date>.log
IBM AD Analyze Client	<Local Workspace>\.metadata\.ez\log\	• ez.log • ez.<date>.log
	<Local Workspace>\.metadata\	.log

Cleaning up the OrientDB database

The OrientDB Maintenance script can be used to clean up the OrientDB database. The script identifies the OrientDB databases that are not longer referenced by the symbolic links and deletes them. The `OrientDBDatabaseMaintenance-x.x.bat` script is located in the `<IBM ADDI Installation Folder>\support\OrientDBDatabaseMaintenance\` folder.

Note: For security reasons, the deletion steps are commented (REM) at lines 255 and 257.

1. Go to `<IBM ADDI Installation Folder>\support\OrientDBDatabaseMaintenance\` folder and run the `OrientDBDatabaseMaintenance-x.x.bat` script.
2. If unnecessary database folders are found, use option `Y (Yes)` to remove all unused database folders.
3. Check the logs under the `C:\ADlogs\BatchServer\orientdb\MaintenanceLogs\` folder to review any database folder name that was removed.

All unused database folders are removed.

Disaster Recovery

Backing Up Steps for Components

I. AD Configuration Server

Note:

- Before you start to back up on Windows, make sure that the IBMApplicationDiscoveryConfigurationAdminService and IBMApplicationDiscoveryConfigurationService services are stopped.
- Before you start to back up on Linux, make sure that the `startServer.sh` and `startWebServerUI.sh` files, corresponding to AD Configuration Server, are not running.

Procedure

1. Back up folder *<IBM AD Configuration Server Installation Path>*\store. The database and data of Configuration Server are stored in the store folder.
2. Back up folder *<IBM AD Configuration Server Installation Path>*\conf. The configuration files are stored in the conf folder.

II. AD Build Client

Procedure

1. Back up AD Build projects.
   a. Back up project folders.
      i) Include the default path for AD Build projects in the backing up procedure. The default path is defined in the **Default projects path** field. To see the value, click the **AD Build Client** tab in IBM AD Configuration Server. See the following figure for illustration:



*Figure 72. **Default projects path** field*

      ii) If project folders exist in the paths that are not the default path for AD Build projects, make sure to include those paths in the backing up procedure. To identify the paths, run the following query on the SQL Server where AD Build projects exist:

```
SET NOCOUNT ON;
DECLARE @t TABLE (DB VARCHAR(100), PathStrFull VARCHAR(1000)) ;
DECLARE @SQL NVARCHAR(MAX);

SELECT @SQL = STUFF((
    SELECT CHAR(13) + 'SELECT ''' + name + ''', PropValue FROM [' + name + '].[dbo].[Pj_ProjectProp] WHERE
ID_PropType = 27'
    FROM sys.databases
    WHERE OBJECT_ID('[' + name + '].[dbo].[Pj_ProjectProp]') IS NOT NULL
    FOR XML PATH(''), TYPE).value('.', 'NVARCHAR(MAX)'), 1, 1, '');

INSERT INTO @t(DB, PathStrFull)
EXEC sys.sp_executesql @SQL ;

SELECT ProjectsPaths
FROM (
       SELECT DISTINCT
       CASE WHEN CHARINDEX('EZ_', DB) = 1 THEN SUBSTRING(PathStrFull, 1, PATINDEX( '%[\]' + SUBSTRING(DB, 4,
LEN(DB) - 3) + '[\]%', PathStrFull))
            ELSE NULL
        END AS ProjectsPaths
       FROM @t t
      ) src
WHERE LEN(src.ProjectsPaths) > 0 ;
```

   b. Back up the relational databases of AD Build Client projects.

      Every AD Build Client project has its own corresponding relational database. Include all the relational databases with names that start with string "EZ_" in the backing up procedure, and make sure that they can be restored at any time. See the following figure for illustration:

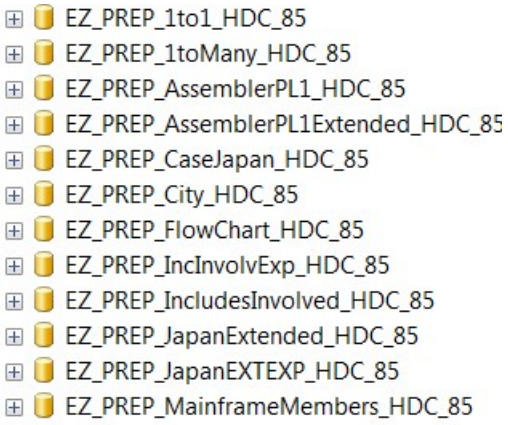


Figure 73. Relational database examples

2. Back up the source code files that are loaded in AD Build Client.

   a. Include the path where source code files are automatically downloaded from a Mainframe system using AD Connect for Mainframe and stored in the backing up procedure. This path is defined in the **Path for the retrieved members** field. To see the value, click **Install Configurations > IBM Application Discovery Build Client** in IBM AD Configuration Server. See the following figure for illustration:

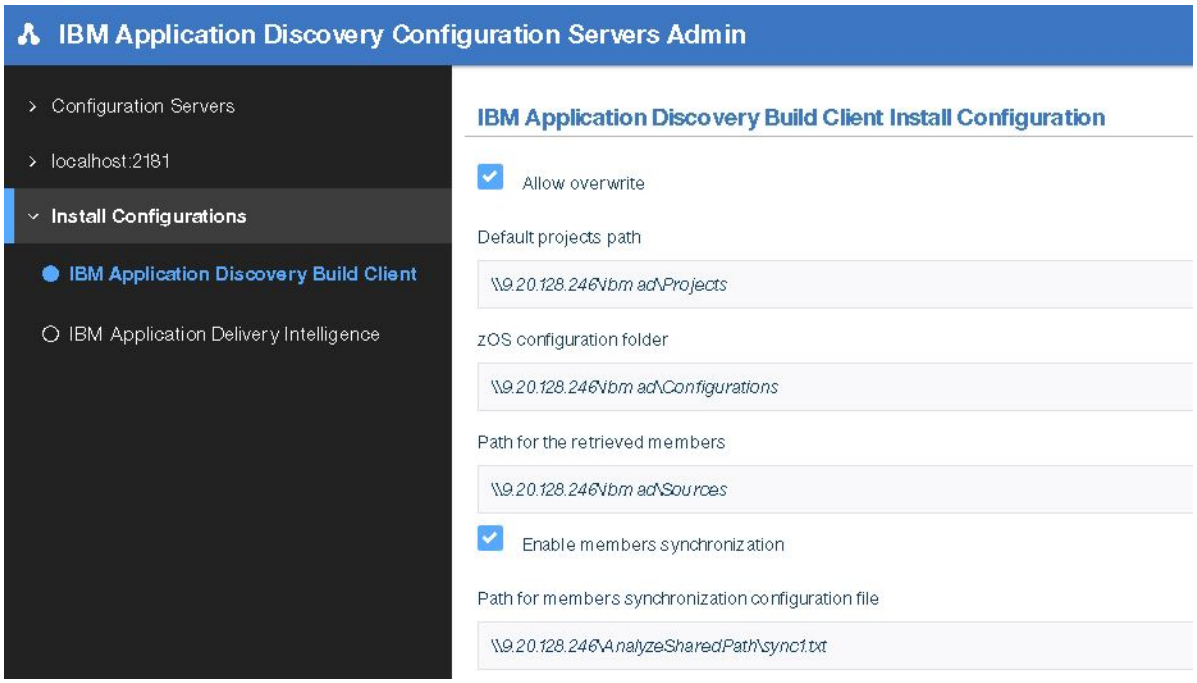


*Figure 74. **Path for the retrieved members** field*

   b. For the source code files that are manually added or loaded in AD Build projects, make sure to include the paths where those files exist in the backing up procedure.

3. Back up AD Build configurations.

   a. Back up z/OS configurations. The folder where z/OS configurations are stored is defined in the **zOS configuration folder** field. To see the value, click **Install Configurations > IBM Application Discovery Build Client** in IBM AD Configuration Server. See the following figure for illustration:

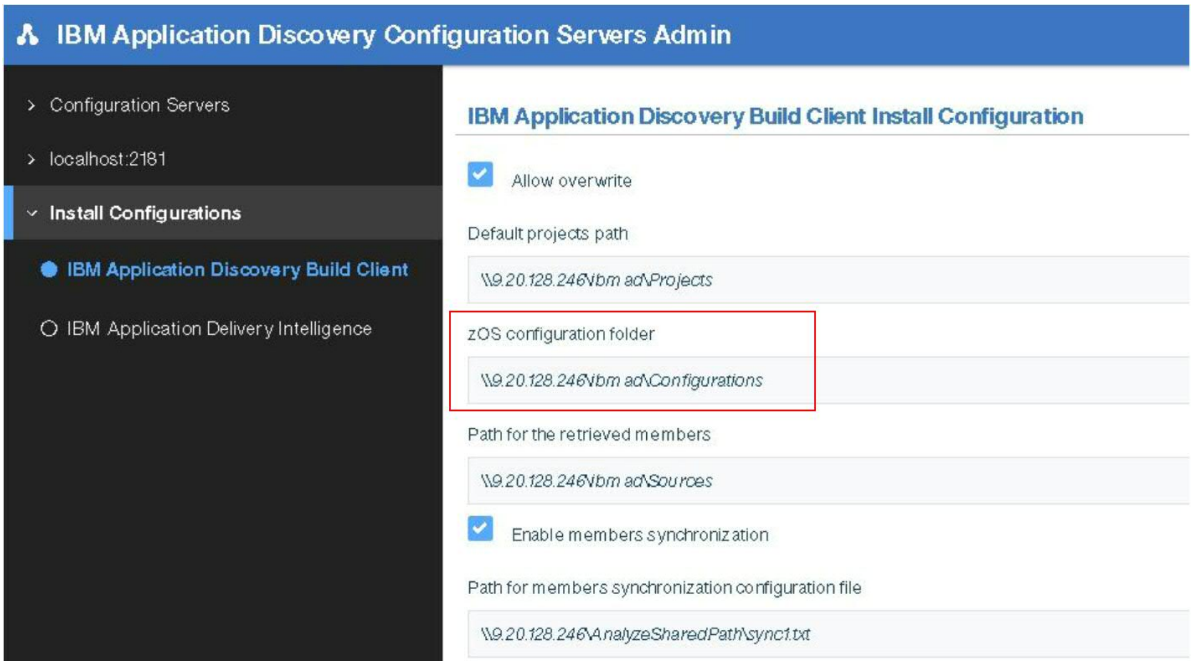


*Figure 75. **zOS configuration folder** field*

b. Back up the Synchronization configuration file. This file is used for the mainframe members synchronization process, and is defined in the **Path for members synchronization configuration file** field. To see the value, click **Install Configurations > IBM Application Discovery Build Client** in IBM AD Configuration Server. See the following figure for illustration:

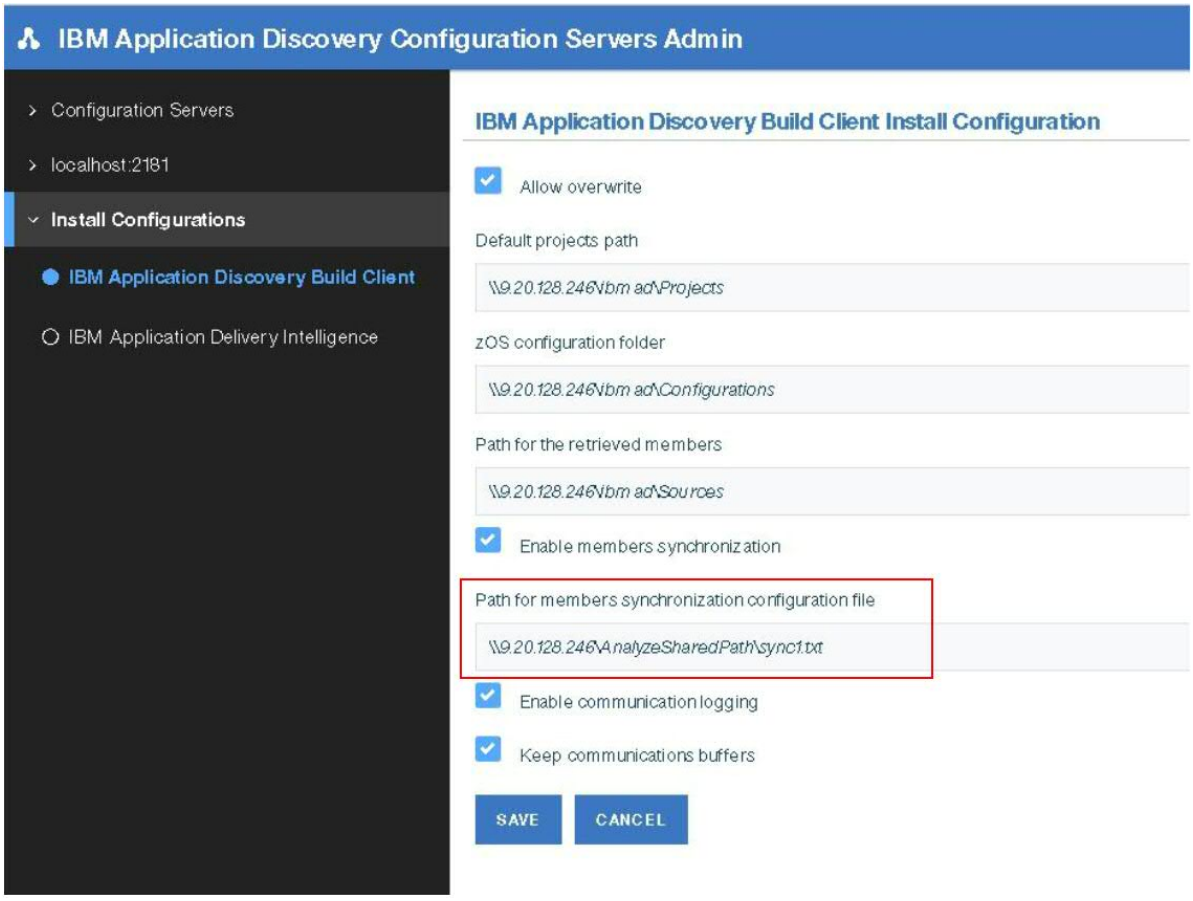


*Figure 76. **Path for members synchronization configuration file** field*

c. For environments that download assets from Endevor via IBM AD Connect for Mainframe, back up the following Endevor configuration files:

- Promotion routes configuration file
- Types list configuration file

The two files are configured for Endevor source code download. To see the file paths, click the **ENDEVOR Info** tab in the "zOS configuration" window. See the following figure for illustration:

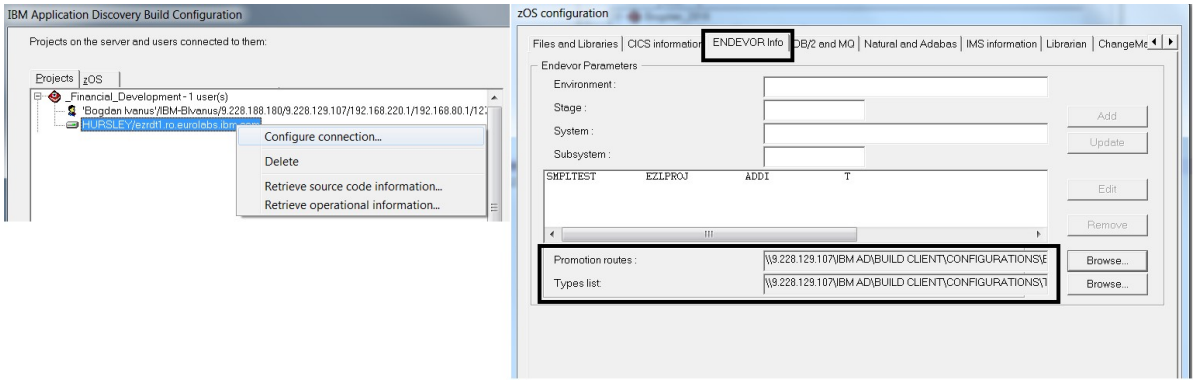


*Figure 77. **ENDEVOR Info** tab*

4. Back up the following AD Validation Service configuration files. They are stored in the *<IBM AD Build Client InstallationFolder>\IBMApplicationDiscoveryValidationServer* folder.
   - `ApprovalRequestParameters.txt`
   - `CompletionCodeVsMessage.txt`
   - `FoldersMapping.txt`
   - `IncludesOrder.txt`
   - `LoopbackResult.txt`
   - `ParallelValidationParameters.txt`
   - `ProjectsMapping.txt`
   - `ProjectsMappingParallelBuild.txt`
   - `ServicePort.txt`
5. Back up the *<IBM AD Build Client InstallationFolder>\IBMApplicationDiscoveryValidationServer\ReportsGenerator* folder. By default, the `conf`, `data`, and `tmp` folders are contained by the `ReportsGenerator` folder. If the paths for the three folders are changed in the `server.properties` configuration file, back up accordingly.
6. If the Rule Based component is used, back up Rule Based reports and queries.
   a. Back up Rule Based reports.

   By default, the rule-based data is generated and stored in the `data` folder, which is already included in the backing up procedure. For more information about the `data` folder, see Step 5.

   If the default folders that are related to the Rule Based component are changed in the `project.properties` file, back up the folders as configured. If folders for specific projects are defined, back up the corresponding folders as well.

```
## output folder for rule based reports generated by source based rules. Default, data
folder
ruleBased.reportsFolder=

## output folder for csv files generated by data based rules. Default, data folder
#ruleBased.csvFolder=

## output folder for controlTotals files generated by source based rules. Default, data
folder
## used only if generateTotals is true
```

```
#ruleBased.totalsFolder=

## output folder for rule based reports generated by source based rules. Default, data
folder
#project.<projectName>.ruleBased.reportsFolder=

## output folder for csv files generated by data based rules. Default, data folder
#project.<projectName>.ruleBased.csvFolder=

## output folder for controlTotals files generated by source based rules. Default, data
folder
## used only if generateTotals is true
#project.<projectName>.ruleBased.totalsFolder=
```

b. Back up Rule Based queries.

If you use Rule Based global settings, check the paths where input and queries are stored in the `ruleBasedConfig.properties` file.

If you use Rule Based project settings, check the paths where input and queries are stored in the rule properties file of each project. Also, if the rule properties files are not stored in the `conf` folder, back up the folders where they are stored.

III. AD Batch Server

Note:

- Before you start to back up on Windows, make sure that the IBMApplicationDiscoveryBatchService, IBMApplicationDiscoveryGraphDBService, and IBMApplicationDiscoveryWebService services are stopped.
- Before you start to back up on Linux, make sure that the `server.sh` (corresponding to GraphDB), `startServer.sh` (corresponding to AD Batch Server), and `startWBServer.sh` (corresponding to AD Web Service) files are not running.
- Generally, the `data` and `conf` folders that are distributed by the AD Batch Server installation must be included in the backing up procedure. See the following detailed description for the folders.

```
## path to the directory where configuration files are placed
## default: ${install.dir}/conf
conf.dir=<path>\conf

## path to the directory where data files are placed
## default: ${install.dir}/data
data.dir=<path>\data

## path to the directory where temporary data is placed
## default: ${data.dir}/tmp
tmp.dir=<path>\tmp
```

If the default paths for the `data`, `conf`, and `tmp` folders are changed in the `server.properties` configuration file, back up accordingly.

Procedure

1. Back up configurations.
   a. Back up the *<Batch Server Installation Path>*\Conf configuration folder.
   b. Back up the following configuration folders from the `tmp` folder:
      - *<Batch Server Installation Path>**data*\tmp\mfp
      - *<Batch Server Installation Path>**data*\tmp\ver
2. Back up OrientDB databases (graph databases).

   Back up the default folder *<Batch Server Installation Path>**data*\tmp\gdb. The size of this folder might be large, depending on the number and size of AD Build Client projects.
3. Back up the index folders as defined in the `project.properties` configuration file. The indexes that are generated by AD Batch Server and used for Search In Files analysis in AD Analyze Client are stored

in these folders. If indexes for specific projects are defined, back up the corresponding index folders as well.

```
## index global settings
## output folder where indexes are stored. (project name will be added by default)
## this setting does not override database entry!
## Note that index folder should be a shared folder in order for all Analyze client to be able to use it
index.indexFolder=\\9.20.128.211\\IBM AD\\Batch Server\\Indexes

### index project settings
## output folder where index for this project is stored. must be unique per project
## this setting does not override database entry!
project.
project.<projectName>.index.indexFolder==\\9.20.128.211\\IBM AD\\Batch Server\\Indexes\\<projectName>
```

4. Back up the *<Batch Server Installation Path>**data*\tmp\wsmetrics folder. The Web Services data is stored in this folder.
5. If the Rule Based component is used, back up Rule Based reports and queries.
   a. Back up Rule Based reports.

      By default, the rule-based data is generated and stored in the data folder, which is already included in the backing up procedure. For more information about the data folder, see the preceding Note.

      If the default folders that are related to the Rule Based component are changed in the project.properties file, back up the folders as configured. If folders for specific projects are defined, back up the corresponding folders as well.

      ```
      ## output folder for rule based reports generated by source based rules. Default, data
      folder
      ruleBased.reportsFolder=

      ## output folder for csv files generated by data based rules. Default, data folder
      #ruleBased.csvFolder=

      ## output folder for controlTotals files generated by source based rules. Default, data
      folder
      ## used only if generateTotals is true
      #ruleBased.totalsFolder=

      ## output folder for rule based reports generated by source based rules. Default, data
      folder
      #project.<projectName>.ruleBased.reportsFolder=

      ## output folder for csv files generated by data based rules. Default, data folder
      #project.<projectName>.ruleBased.csvFolder=

      ## output folder for controlTotals files generated by source based rules. Default, data
      folder
      ## used only if generateTotals is true
      #project.<projectName>.ruleBased.totalsFolder=
      ```

   b. Back up Rule Based queries.

      If you use Rule Based global settings, check the paths where input and queries are stored in the ruleBasedConfig.properties file.

      If you use Rule Based project settings, check the paths where input and queries are stored in the rule properties file of each project. Also, if the rule properties files are not stored in the conf folder, back up the folders where they are stored.
6. Back up the annotation database.

   In earlier versions of IBM AD, IBM AD Batch Server creates a database that is called EZ#Annotations in Microsoft SQL Server. This database is created on the SQL Server as defined in the server.properties configuration file.

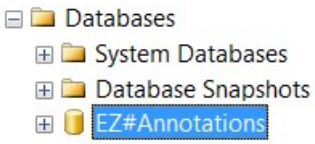


Figure 78. EZ#Annotations database

Starting with IBM AD V5.1.0, IBM AD Batch Server reads the configurations that are made in **IBM AD Configuration Sever**, under **Annotations Database** and create the annotations database in the relational database server that is specified there, by using the database name and the schema set by the user.

You can still add EZ#Annotations in database name field, together with dbo schema and associate the relational database server in which the annotation database was created, the one defined in the `server.properties` configuration file.

Using the configurations set in **Annotations Database** configurations area you can find out the name of your annotations database, the schema and the associated relational database server in order to know what database to back up.

IV. AD Analyze Server

Note:

- Before you start to back up on Windows, make sure that the IBMApplicationDiscoveryAnalyzeService service is stopped.
- Before you start to back up on Linux, make sure that the `StartServer.sh` file, corresponding to IBM AD Analyze Server, is not running.

Procedure

If the IBM AD Analyze Server component is used, follow the steps to back up:

1. Back up the IBM AD Analyze Server database.

   IBM AD Analyze Server creates a database in Microsoft SQL Server. To identify the IBM AD Analyze Server database, start IBM AD Analyze Server, and click the **Database settings** tab. See the following figure for the illustration of the details about the SQL Server location and the database:

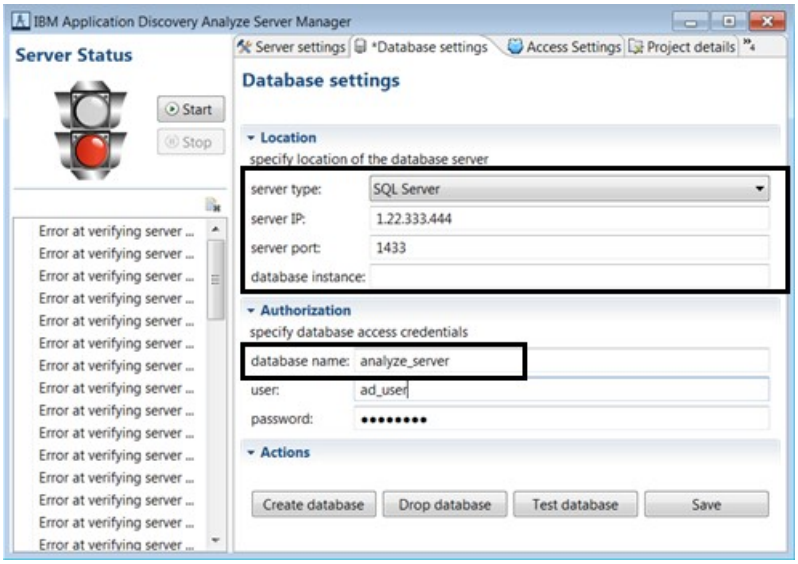


Figure 79. IBM AD Analyze Server database details

2. Back up the following configuration files. These three files are required for IBM AD Analyze Server Service, and they are stored in folder *<IBM AD Analyze Server Installation Folder>*.

- server.properties
- client.properties
- database.properties

V. AD Analyze Client

Procedure

Back up the Analyze Client workspace. All information about configuration settings is stored in the Analyze Client workspace. To see the workspace location, click **Windows > Preferences > Workspace**.

VI. AD Web Services

Note:

- The IBM AD Catalog, IBM AD Audit, and IBM AD BRD web services are deployed on the same IBM WAS Liberty Web Server. Back up the web services together because they are using the same configuration file.
- The server.xml file is the only file that is changed and can be configured after the deployment of the IBM AD Catalog, IBM AD Audit, and IBM AD BRD web services.

Procedure

1. Back up the IBM AD Catalog and IBM AD Audit databases.

   Both IBM AD Catalog and IBM AD Audit require a database. To back up the databases that are used by the IBM AD Catalog and IBM AD Audit, see the server.xml file and get the database connection details.

   - For the Audit web service, search for the following data source:

     ```
     <dataSource id="DefaultDataSource" jndiName="jdbc/datasource" type="javax.sql.DataSource">
     ```

   - For the Catalog web service, search for the following data source:

     ```
     <dataSource id="ADCatRDB" jndiName="jdbc/ad/catalog/relational"
     type="javax.sql.DataSource">
     ```

2. Back up the server.xml configuration file.

   Add the server.xml file in the backup directory for web services. The server.xml file is under the web service installation path: *<was_liberty_path>*\usr\servers\ad_server.

3. For the IBM AD Catalog web service, back up the DataColector configuration files DC.properties and zoscdc.cmd. The files are stored under the zoscDataCollector directory in the same location as the IBM WAS Liberty.
4. For the IBM AD BRD web service, back up the conf.brd-ws folder that includes the application.properties file.

VII. AD Services

Procedures

Back up the configuration files for the following **IBM AD Services** as follows:

- **Authentication Server (DEX)**

  Back up the dex.yaml file that is stored in the <IBM ADDI Installation Folder>/Authentication Server (DEX)/conf/ folder.

- **IBM AD File Service**

Back up the `conf.yaml` file that is stored in the `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/conf/` folder.

- **IBM AD Search Service**

  Back up the `conf.yaml` file that is stored in the `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/conf/` folder.

- **IBM AD Manual Resolutions Service**

  1. Back up the `conf.yaml` file that is stored in the `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions Service/conf/` folder.
  2. Back up the folder where the journals were created by following the same path that is set in the `conf.yaml`, both the default path and the overridden path. The values of the **projectPath** parameter and the **projects** section present in the `conf.yaml` need to be taken into consideration.

- **IBM AD Mainframe Projects Service**

  Back up the `conf.yaml` file that is stored in the `<IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/conf/` folder.

- **IBM AD Cross Applications Service**

  Back up the `conf.yaml` file that is stored in the `<IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/conf/` folder.

Restoring Steps for Components

Important:

- All the following components must be installed, and their corresponding services (Windows) and processes (Linux) must be stopped before you start to restore.
- Make sure that you restore in the same order as mentioned in this document. Restoring in a different order might make the components not work properly because the components have dependencies.
- Make sure to maintain the same IP/Hostname values for the AD components and AD SQL Server machines when you restore, considering that AD components are using IP/Hostname values for some of their configurations.

I. AD Configuration Server

Note:

- Before you start to restore on Windows, make sure that the IBMApplicationDiscoveryConfigurationAdminService and IBMApplicationDiscoveryConfigurationService services are stopped.
- Before you start to restore on Linux, make sure that the `startServer.sh` and `startWebServerUI.sh` files, corresponding to AD Configuration Server, are not running.

Procedure

1. Restore the database and data of Configuration Server. For more information about the database and data, see step 1 in section backing up steps for AD Configuration Server.
2. Restore the configuration files. For more information about the files, see step 2 in section backing up steps for AD Configuration Server.

   Note: At this point, IBM AD Configuration Server is configured and up and running.

II. AD Build Client

Note: Before you start to restore AD Build Client, AD Configuration Server must be restored, and AD Build Client is highly recommended to be installed.

Procedure

1. Restore AD Build projects.
   a. Restore the project folders. For more information about the folders, see substep a of step 1 in section backing up steps for AD Build Client.

      Note: Make sure to keep the same security/share level for each folder as they were defined.

   b. Restore the relational databases of AD Build Client projects. For more information about the databases, see substep b of step 1 in section backing up steps for AD Build Client.

      Note: Make sure to maintain the same ownership for each database as they were defined, and that the owner has all the necessary rights as defined during installation and configuration.
2. Restore the source code files that are loaded in AD Build Client. For more information about the files, see step 2 in section backing up steps for AD Build Client.

   Note: Make sure to keep the same security/share level for each folder as they were defined.
3. Restore AD Build configurations. For more information about the configurations, see step 3 in section backing up steps for AD Build Client.

   Note:

   - Make sure to keep the same security/share level for each folder as they were defined.
   - At this point, IBM AD Build Client is configured and up and running, and all the projects are available and can be used.
4. If AD Validation Service is used, restore AD Validation Service configuration files. For more information about the files, see step 4 and 5 in section backing up steps for AD Build Client.
5. If the Rule Based component is used, restore Rule Based reports and queries. For more information about the reports and queries, see step 6 in section backing up steps for AD Build Client.

   Note: At this point, IBM AD Validation Service is configured, and the corresponding service can be started.

III. AD Batch Server

Note:

- Before you start to restore on Windows, make sure that the IBMApplicationDiscoveryBatchService, IBMApplicationDiscoveryGraphDBService, and IBMApplicationDiscoveryWebService services are stopped.
- Before you start to restore on Linux, make sure that the `server.sh` (corresponding to GraphDB), `startServer.sh` (corresponding to AD Batch Server), and `startWBServer.sh` (corresponding to AD Web Service) files are not running.
- Before you start to restore AD Batch Server, AD Configuration Server and AD Build Client must be restored or up and running.

Procedure

1. Restore the configuration folders. For more information about the folders, see step 1 in section backing up steps for AD Batch Server.
2. Restore OrientDB databases (graph databases).
   a. Restore the default folder. For more information about the folder, see step 2 in section backing up steps for AD Batch Server.
   b. To re-create the symbolic links, go to `<IBM AD Batch Server Installation folder>` and run `recoverGDBSymbolicLinks.bat` on Windows and `recoverGDBSymbolicLinks.sh` on Linux.

      Note: Both of the files must be executed with the following two parameter values:

Location of the graph databases
"<IBM AD Batch Server Installation path>\data\tmp\gdb"

Location where the symbolic links must be created
"<IBM AD Batch Server Installation path>\orientdb\orientdb-community\databases"

Example: `recoverGDBSymbolicLinks.bat "<IBM AD Batch Server Installation path>\data\tmp\gdb" "<IBM AD Batch Server Installation path>\orientdb\orientdb-community\databases"`

3. Restore the index folders. For more information about the folders, see step 3 in section backing up steps for AD Batch Server.

   Note: Make sure to keep the same security/share level for each folder as they were defined.
4. Restore the Web Services data. For more information about the data, see step 4 in section backing up steps for AD Batch Server.
5. If the Rules Based component is used, restore Rules Based reports. For more information about the reports, see step 5 in section backing up steps for AD Batch Server.
6. Restore the annotation database: EZ#Annotations. For more information about the database, see step 6 in section backing up steps for AD Batch Server.

   Note:

   - Make sure to maintain the same ownership for the annotation database as it was defined, and that the owner has all the necessary rights as defined during installation and configuration.
   - At this point, IBM AD Batch Server is configured, and all the corresponding services (Windows) and processes (Linux) can be started.

IV. AD Analyze Server

Note:

- Before you start to restore on Windows, make sure that the IBMApplicationDiscoveryAnalyzeService service is stopped.
- Before you start to restore on Linux, make sure that the `StartServer.sh` file, corresponding to Analyze Server, is not running.

Procedure

1. Restore the Analyze Server database. For more information about the database, see step 1 in section backing up steps for AD Analyze Server.

   Note: Make sure to maintain the same ownership for the Analyze Server database as it was defined, and that the owner has all the necessary rights as defined during installation and configuration.
2. Restore the Analyze Server configuration files. For more information about the files, see step 2 in section backing up steps for AD Analyze Server.

   Note: At this point, IBM AD Analyze Server is configured, and all the corresponding services (Windows) and processes (Linux) can be started.

V. AD Analyze Client

Procedure

1. Restore the Analyze Client workspace. For more information about the workspace, see the procedure in section backing up steps for AD Analyze Client.
2. Set the Analyze Client Eclipse to use the backup workspace.

   a. Open the Analyze Client instance.

b. Click **File > Switch Workspace > Other...**.

c. Select the backup workspace.

Note: At this point, IBM AD Analyze Client is configured and ready to use for analysis.

VI. AD Web Services

Procedure

1. Restore the IBM AD Catalog and IBM AD Audit databases. For more information about the databases, see step 1 in section backing up steps for AD Web Services.

   Note: Make sure to maintain the same ownership for each database as it was defined, and that the owner has all the necessary rights as defined during installation and configuration.

2. Restore the `server.xml` configuration file. For more information about the file, see step 2 in section backing up steps for AD Web Services.

   If the database connection details were changed, edit the `server.xml` file to point to the correct location of the SQL/Db2 server, and change the credentials accordingly.

3. For IBM AD Catalog, restore the DataCollector configuration files `DC.properties` and `zoscdc.cmd`. For more information about the files, see step 3 in section backing up steps for AD Web Services.

   Create a directory *<was_liberty_path>*`\zoscDataCollector`, and put the DataCollector configuration files in it.

4. For IBM AD BRD, restore the `conf.brd-ws` folder which includes the `application.properties` file.

   Note: At this point, IBM AD Catalog, IBM AD Audit and IBM AD BRD are configured, and all their services (Windows) and processes (Linux) can be started.

VII. AD Services

Procedures

Restore the configuration files for the following **IBM AD Services** as follows:

- **Authentication Server (DEX)**

  Restore the `dex.yaml` file and place it under the `<IBM ADDI Installation Folder>/Authentication Server (DEX)/conf/` folder.

- **IBM AD File Service**

  Restore the `conf.yaml` file and place it under the `<IBM ADDI Installation Folder>/IBM Application Discovery File Service/conf/` folder.

- **IBM AD Search Service**

  Restore the `conf.yaml` file and place it under the `<IBM ADDI Installation Folder>/IBM Application Discovery Search Service/conf/` folder.

- **IBM AD Manual Resolutions Service**

  1. Restore the `conf.yaml` file and place it under the `<IBM ADDI Installation Folder>/IBM Application Discovery Manual Resolutions/conf/` folder.
  2. Restore the folders where the journals were created by following the same path that is set in the `conf.yaml` file.

     Note: If the path to the folders where the journals are placed is different from the previous one, make sure that you add the new path in the `conf.yaml` configuration file.

- **IBM AD Mainframe Projects Service**

Restore the conf.yaml file and place it under the <IBM ADDI Installation Folder>/IBM Application Discovery Mainframe Projects Service/conf/ folder.

- **IBM AD Cross Applications Service**

  Restore the conf.yaml file and place it under the <IBM ADDI Installation Folder>/IBM Application Discovery Cross Applications Service/conf/ folder.

Chapter 13. Uninstalling IBM AD

Important:

- Before starting the uninstall process, make sure that all IBM AD Build and IBM AD Analyze clients are closed.
- Uninstall the products in the exact order in which they are presented below.
- As a general note, understand that **Force the Deletion** option deletes all the contents of a specific folder, meaning that configurations are lost.
- After **IBM AD Build Client** is uninstalled **make sure to reboot** the workstation.
- In case you are upgrading and need to uninstall first, make sure to back up the configuration information. For more details on upgrading, go to Chapter 4, “Upgrading,” on page 31.

1. Uninstall IBM AD:

   Under Windows

   - Stop all services that are related to IBM AD Batch Server, IBM AD Configuration Server, and IBM AD Analyze Server:
     - IBM Application Discovery Analyze Service
     - IBM Application Discovery Batch Service
     - IBM Application Discovery Configuration Admin Service
     - IBM Application Discovery Configuration Service
     - IBM Application Discovery GraphDB Service
   - Click **Start** and then select **All Programs** > **IBM Application Discovery and Delivery Intelligence** > **Unistall IBM Application Discovery and Delivery Intelligence**.
   - **Alternative CLI uninstall**: open a command line in *Installation Path*\IBM Application Discovery and Delivery Intelligence\Uninstall and run the following command:

     ```
     java –jar uninstaller.jar -cki
     ```

   Under Linux/zLinux

   - Stop all the processes that are related to IBM AD Batch Server, IBM AD Configuration Server, and IBM AD Analyze Server.
   - Stop all Java™ processes related to IBM AD Batch Server, IBM AD Configuration Server, and IBM AD Analyze Server.
   - Go to the installation folder (By default, the installation folder is *Installation Path*\IBM Application Discovery and Delivery Intelligence\Uninstall), execute uninstaller.sh and follow the uninstalling steps.
   - **Alternative CLI uninstall**: open a command line in *Installation Path*\IBM Application Discovery and Delivery Intelligence\Uninstall and run the following command:

     ```
     ./uninstaller.sh -cki
     ```

2. Uninstall IBM AD Analyze Client: To uninstall IBM AD Analyze Client: Go to **Eclipse client** > **Help** > **About Eclipse SDK**. In the **About Eclipse SDK** dialog window: Click **Installation Details**. In the Eclipse SDK Installation Details dialog window: From the **Installed Software** tab, select all the components that start with **Application Discovery** then click **Uninstall**. The **Uninstall** dialog window presents the list of components that will be uninstalled: Click **Finish** to start the uninstall process.

© Copyright IBM Corp. 2010, 2022

Documentation Notices

© Copyright International Business Machines Corporation 2010, 2020. US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

```
IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US
```

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
Intellectual Property Licensing
Legal and Intellectual Property
Law
IBM Japan Ltd.
19-21, Nihonbashi-
Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan
```

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary. This information is for planning purposes only.

The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE: This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.



SC28-3160-04

