IBM Application Discovery for IBM Z Connect for Mainframe V5.1.0

Configuration Guide

GC27-8974-06

Contents

Chapter 1. Accessibility Features for IBM Application Discovery for IBM Z

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.

The IBM Application Discovery for IBM Z online product documentation in IBM Knowledge Center is enabled for accessibility. The accessibility features of IBM Knowledge Center are described in the Accessibility section of the IBM Knowledge Center help (https://www.ibm.com/support/knowledgecenter/en/about/releasenotes.html).

Keyboard navigation

This product uses standard navigation keys.

Interface information

For alternative installation using Command Line Installation (CLI), refer to section Alternative Installation for ADDI Using CLI in *IBM AD Installation and Configuration Guide*.

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).

Chapter 2. Introduction

About IBM AD Connect for Mainframe

IBM Application Discovery for IBM Z (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, the IBM AD Connect for Mainframe product is being distributed in SMP/E format. As a result, the EZL prefix has been changed to IAY. Any instance of EZL should be assumed to be IAY.

Minimum Requirements for Installation

The minimum installation requirements for IBM AD Connect for Mainframe are:

- z/OS® version 2.2 or later.
- The maximum disk storage space is 5 cylinders.

Minimum Version Requirements for Supported Products

- Any supported version of Db2.
- Any supported version of IMS.
- Any supported version of CICS.
- Any supported version of WebSphere® MQ.
- Any supported version of ChangeMan ZMF.
- Any supported version of CA Endevor. For Endevor version 17.0, make sure to have the RO79413 fix applied.
- Any supported version of CA Workload Automation CA 7.

- Any supported version of TWS or IWS.
- IBM Rational Team Concert version 6.0.3 or later

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

Prerequisites Authorization for Running IBM AD Connect for Mainframe

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 9).
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.
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.

Terms and Conventions

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

- `Command` names are printed as shown.
- "Terms and Conventions" on page 4 in this guide are indicated as shown. For page numbers, refer to the Table of Contents at the beginning of this guide.

- `File references` are printed as shown.
- **Button names** and options/functions within a dialog box are printed as shown.

Chapter 3. Installation Media

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.

Files Included in the Connect for Mainframe Distribution

This is part of the initial installation.

The following files are included:

- **IBM.HALT505.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.HALT505.F1.BIN** – a compressed SMP/E REL file, containing all the load modules for **IBM AD Connect for Mainframe**.
- **IBM.HALT505.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.HALT505.F3.BIN** – a compressed SMP/E REL file, containing all the sample JCL and PROCS needed to run **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.

Connect for Mainframe Installation Steps

Beginning with **v5.0.3**, **IBM AD Connect for Mainframe** is distributed in SMP/E format. For this version, please 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.

If you are trying to install an older version of **IBM AD Connect for Mainframe** (any version prior to v5.0.3), please perform the following steps:

1. FTP the XMIT file (`Vnnn.XMIT.LOADLIB`) into a **predefined** sequential file on the mainframe with the following DCB parameters: LRECL 80, BLKSIZE 3120 and RECFM FB.
2. Use `TSO RECEIVE` command to create **IBM AD Connect for Mainframe** load library.
3. FTP the `listener.txt` into a member in a procedure library which will be used as the JCL for the listener started task. The listener PROC name for the started task could be any name according to the site's naming convention.

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 9.
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.

Configuration for IBM AD Build Configuration

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.

Chapter 4. Technical Configuration Process

Configuring the Listener PROC

About this task

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:

Procedure

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. For IBM TWS, 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.

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.SERENA.XMLIN01 <--FOR CHANGEMAN
//* Create as many XMLOUTnn according to the number of MAXTASK parameter
//XMLOUT01 DD DISP=SHR,DSN=IAY.SERENA.XMLOUT02 <--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.

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.
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 DISP=(NEW,CATLG),
//            DSN=IBMUSER.CA7.REPORT12.FILE%TK,     <===Change(8)
//            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

When **IBM AD Connect for Mainframe** is required to access Db2, the following job, which creates the DBRM plan, should be used:

```
//IAYDB2BD JOB (20,FB3),IBMUSER,MSGLEVEL=(1,1),TIME=20,
// CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,REGION=6M
//*
//JOBLIB DD DSN=DSN910.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(DB9G) <=== DB2 Subsystem name
BIND PLAN(IAYSRC) +
```

```
MEM(IAYDB2DB, IAYDB2VL, IAYDB2TB, IAYDB2TS, IAYDB2PL, IAYDB2CL, +
    IAYDB2SG, IAYDB2PR, IAYDB2IX, IAYDB2KY,IAYDB2TG, IAYDB2VW,  +
    IAYDB2PK,IAYDB2PX, IAYDB2RT, IAYDB200 ) +
CURRENTDATA(NO) ACT(REP) ISO(CS) ENCODING(EBCDIC) +
LIB('IAYV143.MVS110.DBRMLIB.DATA') <=== DBRM lib derived from hlq.SIAYDBRM
END
//SYSIN DD *
GRANT EXECUTE ON PLAN IAYSRC TO PUBLIC;
//*
```

Edit the IAYDB2BD job and modify the following:

1. In the JOBLIB card verify that the library name is set correctly according to your Db2 installation.
2. In the SYSTSIN card, set the DSN SYSTEM name to be your Db2 subsystem name.
3. In case the **loadlib** library name was modified after the transmit, update the LIB entry of the SYSTSIN card accordingly.

For the situation when the members bound into a collection and the plan points to the collection, 2 more additional jobs need to be used.

Create Packages Job:

```
//IAYBNDPK JOB (20,FB3),IBMUSER,MSGLEVEL=(1,1),TIME=20, <== set the correct JOB info
// CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,REGION=6M
//*
//DSN910 JCLLIB ORDER=IAYV210.JCL
//JOBLIB DD DSN=DSN910.SDSNLOAD,DISP=SHR <== DB2 load library
//BINDUNL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB9G) <== DB2 subsystem name
BIND PACKAGE (IAYSRC) OWNER(userdefined) + <== set owner (see all cases of owner in the job)
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2DB) LIBRARY('APPLDISC.DBRM.LIBRARY') <== APPLDISC DBRM library (same goes for all appearance in job)
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2VL) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2TB) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2TS) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2PL) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2CL) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2SG) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2IX) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2KY) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2TG) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2VW) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2PK) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2PX) LIBRARY('APPLDISC.DBRM.LIBRARY')
BIND PACKAGE (IAYSRC) OWNER(userdefined) +
ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) +
MEMBER(IAYDB2RT) LIBRARY('APPLDISC.DBRM.LIBRARY')
END
//
```

Note: The **PACKAGE** name appears also in the second job (see it below) under **PKLIST** and changing the name of the package will affect it.

Plan referring the packages JOB

Important: The owner must have the authority to perform **BIND** on the **PKLIST** specified in this job.

```
//IAYBNDPL JOB (20,FB3),IBMUSER,MSGLEVEL=(1,1),TIME=20, <== set the correct JOB info
// CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,REGION=6M
//*
//DSN910 JCLLIB ORDER=IAYV210.JCL
//JOBLIB DD DSN=DSN910.SDSNLOAD,DISP=SHR
//BINDUNL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
```

```
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB9G) <== DB2 subsystem name
BIND PLAN(IAYTEST1) OWNER(userdefined) + <== set owner (see all cases of owner in the job)
ACTION(ADD) ISOLATION(CS) +
PKLIST(appldisc.*)
END
//
```

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 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 following job should be created whenever accessing **ChangeMan ZMF** via XML service.

The job should 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 wild card substitution for any system.

```
//IAYSERXL JOB (20,FB3),IBMUSER,MSGLEVEL=(1,1),TIME=20,
// CLASS=A,MSGCLASS=Z,NOTIFY=&SYSUID,REGION=6M
```

```
//XML EXEC PGM=SERXMLBC
//STEPLIB DD DISP=SHR,DSN=IAY.SERENA.V7R1M1.CMNZMF.LOAD <== User loadlib
// DD DISP=SHR,DSN=SERENA.SERCOMC.V7R1M1.LOAD <== Serena load library
// DD DISP=SHR,DSN=SERENA.CMNZMF.V7R1M1.LOAD <== Serena load library
// DD DISP=SHR,DSN=IAYV210.LOADLIB <== Application Discovery load library
//SER#PARM DD DISP=SHR,DSN=SERENA.V7R1M1.SERCOMC.TCPIPORT <== Serena TCPIPORT file
//SYSPRINT DD SYSOUT=Z
//SERPRINT DD SYSOUT=Z
//SYSOUT DD SYSOUT=Z
//XMLIN DD DISP=SHR,DSN=IAY.SERENA.XMLIN <=== The input PDS as defined
in the Listener job
//XMLOUT DD DISP=SHR,DSN=IAY.SERENA.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=IAYV210.LOADLIB
//SYSUDUMP DD SYSOUT=*
//INDCB DD *
// ENDIF
```

Note: Make sure the **Changeman** batch jobs are running on the same LPAR were the **z listener** is running.

In addition, create datasets called IAY.SERENA.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 called IAY.SERENA.XMLOUTnn as sequential files with record format VB and record length 5000, where nn goes from 01 to the number of tasks specified in MAXTASK.

The initial track allocation for the above datasets should be based on the formula: Maximum number of members in a ChangeMan library / 80 and another 10% for the additional allocation size.

For example, the biggest **ChangeMan Library** has 44000 members, so the initial allocation would be 44000/80 = 550 plus 55(10%) for the additional allocation size.

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.

Chapter 5. Creating Data Sets with Subsystems Datasets Allocation Panels

About this task

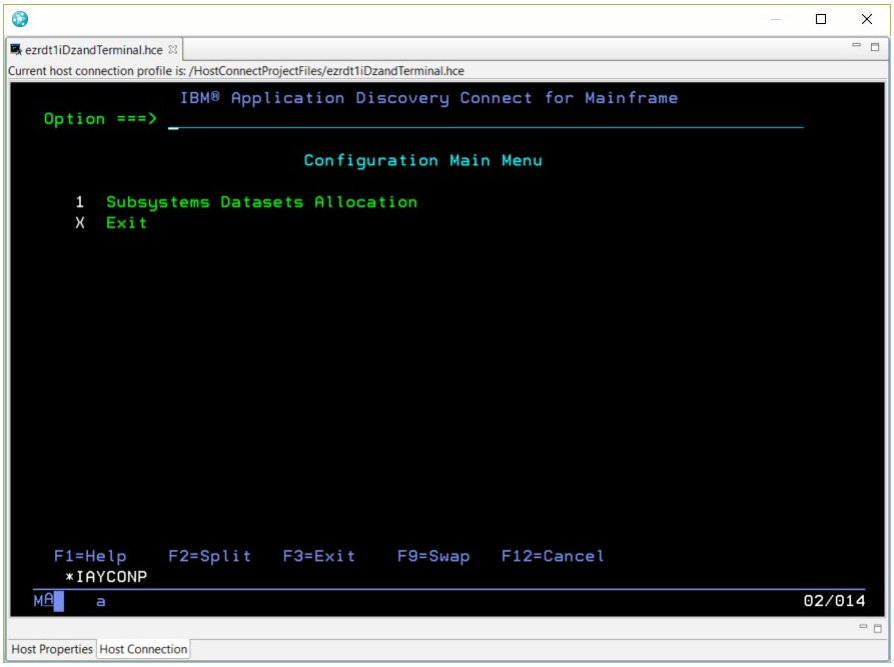
You can create data sets for the ChangeMan, Adabas, and Natural subsystems with **Subsystems Datasets Allocation** panels. All the parameters that are used on the panels are automatically saved in your TSO user ID ISPF profile.

Procedure

1. Go to the **Subsystems Datasets Allocation Main Menu** panel.
   a) Run the following command:

   ```
   TSO EX 'IAYV5102.SAMP.JCL(IAYCONM0)'
   ```

   The **Configuration Main Menu** panel is displayed. You can replace `IAYV5102.SAMP.JCL` with the name of your site-specific library that has the IAYCONM0 member from IBM AD Connect for Mainframe.

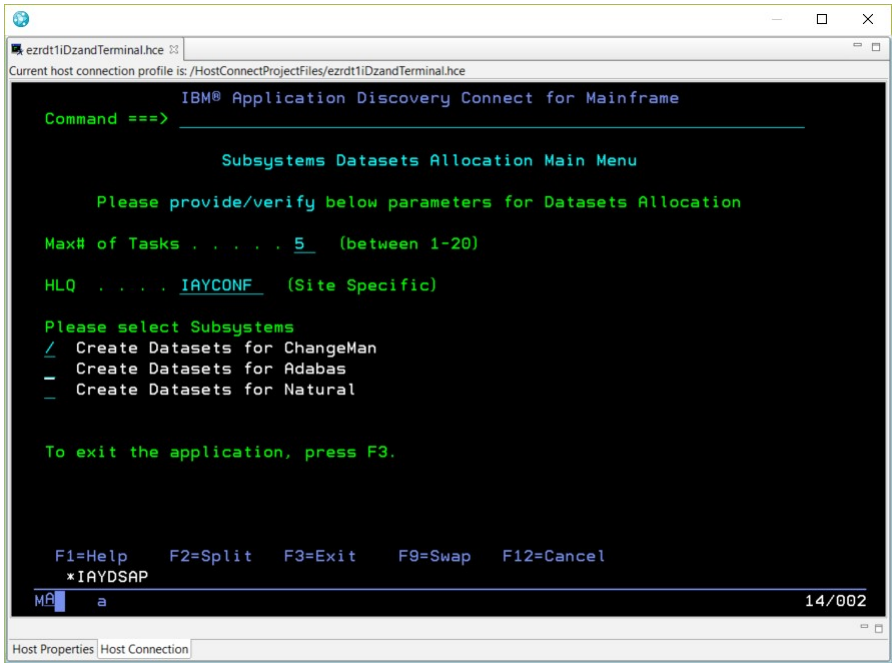


   b) On the **Configuration Main Menu** panel, select option 1.

   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 three subsystems: ChangeMan, Adabas, and Natural. Otherwise, message IAYD883G will be displayed, and you cannot proceed further.

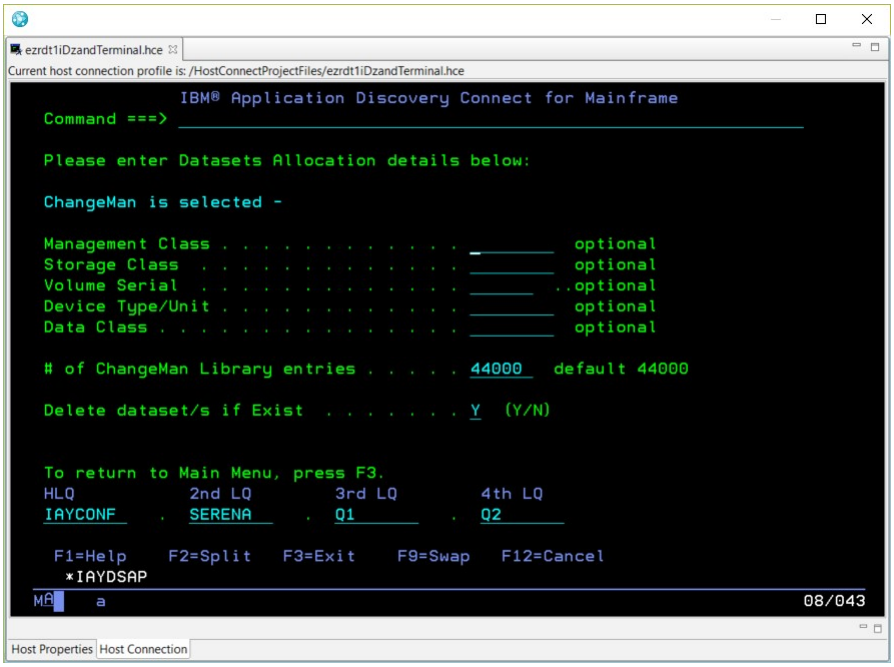


3. Provide or verify the parameters on the data set allocation panel, and then press Enter.
   - If you select Create Datasets for ChangeMan on the **Subsystems Datasets Allocation Main Menu** panel, the following panel is displayed.

     Notes:

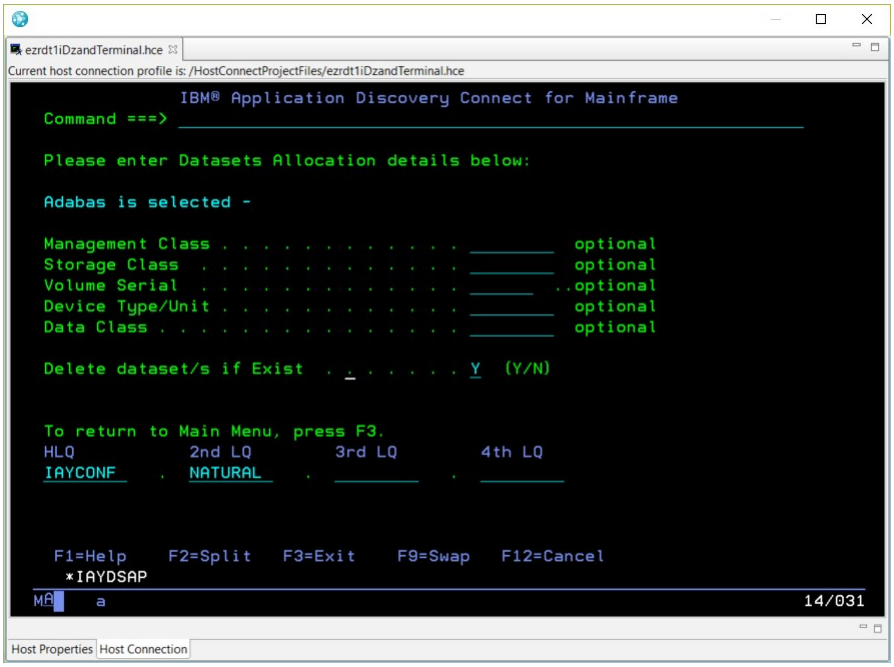
     – By default, the number of ChangeMan library entries is 44000. You can specify another value according to your site specifics in the **# of ChangeMan Library entries** field.

     – 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.



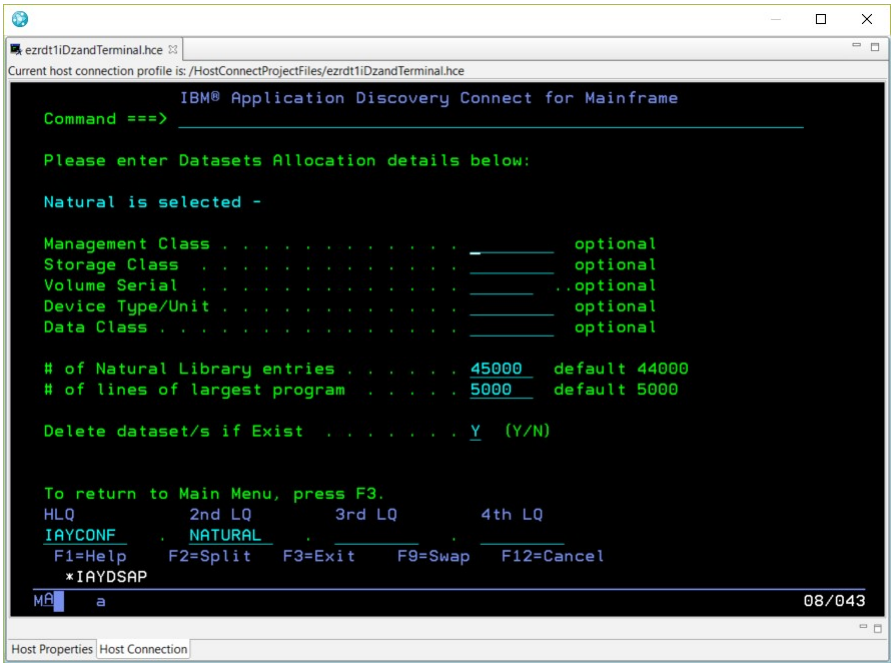
- If you select Create Datasets for Natural on the **Subsystems Datasets Allocation Main Menu** panel, the following panel is displayed.

  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.

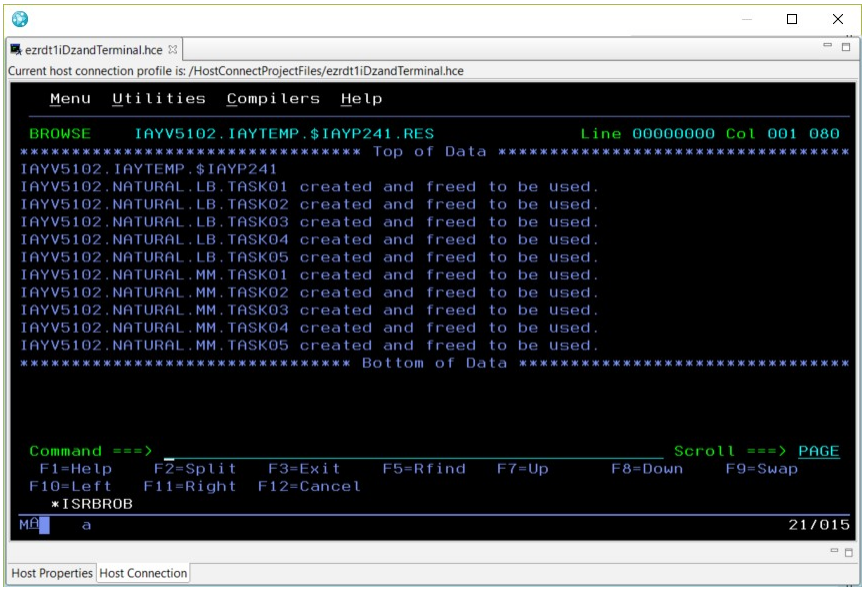
  – By default, the number of the largest program's lines is 5000. You can specify another value according to your site specifics in the **# of lines of largest program** field.

  – The **HLQ** and **2nd LQ** fields must be specified.



Results

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:



What to do next

To go back to the **Subsystems Datasets Allocation Main Menu** panel, press F3.

Chapter 6. RTC to AD Integration

Introduction

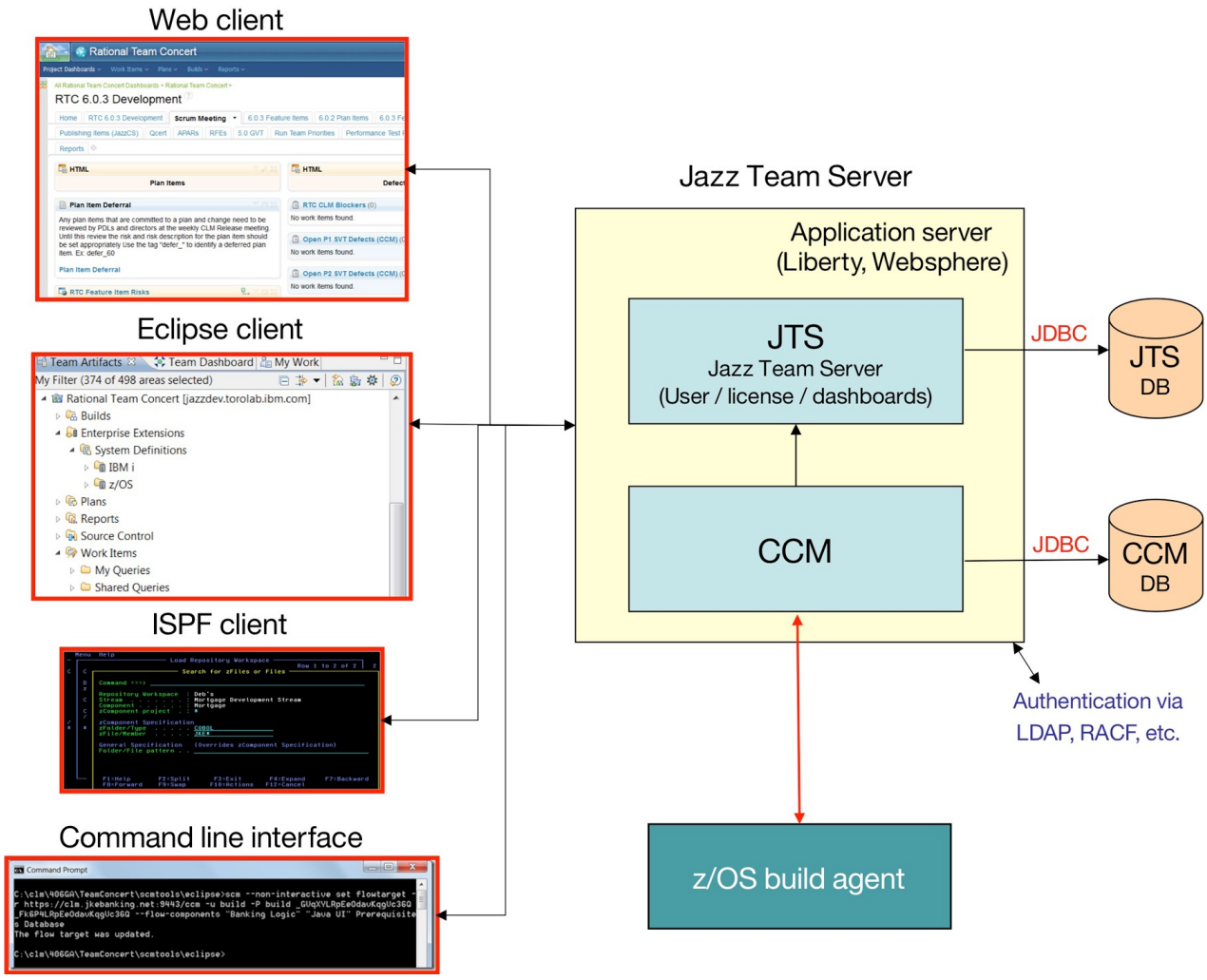
A growing need exists to provide to IBM Application Discovery for IBM Z (IBM AD) users a way to import source files from their Rational® Team Concert® (RTC) source repository into the appropriate AD project sub folders.

Below left, for example, is an RTC 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 RTC 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 RTC source management as it applies to z/OS, and how this tool functions to integrate it with AD source analysis.

RTC Overview

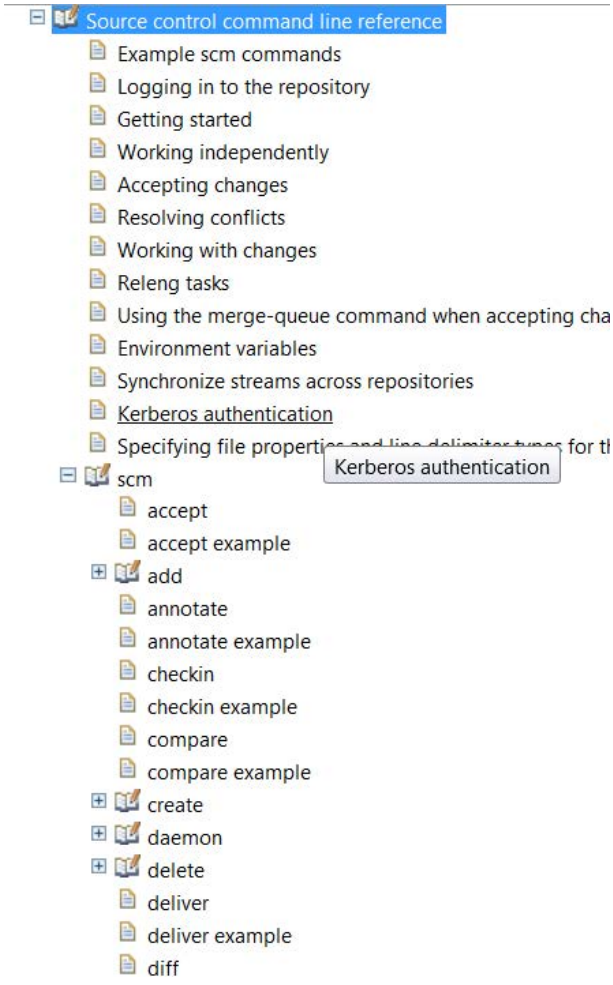
Note: All of the following diagrams, depicting facets of RTC, are copied from the developer works article. For more information, see Migrating to Rational Team Concert in z/OS.



Regarding the preceding diagram, the users need to use one of the RTC client options to work with their source repository stored on a Jazz Team server. The **RTC2AD** tool uses the **RTC command line interface** (CLI), depicted at the lower left corner of the diagram.

On Windows™, the RTC 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 RTC Eclipse Client and so allows scripted interactions with RTC. This tool is the core of the script that provides **RTC2AD** integration.

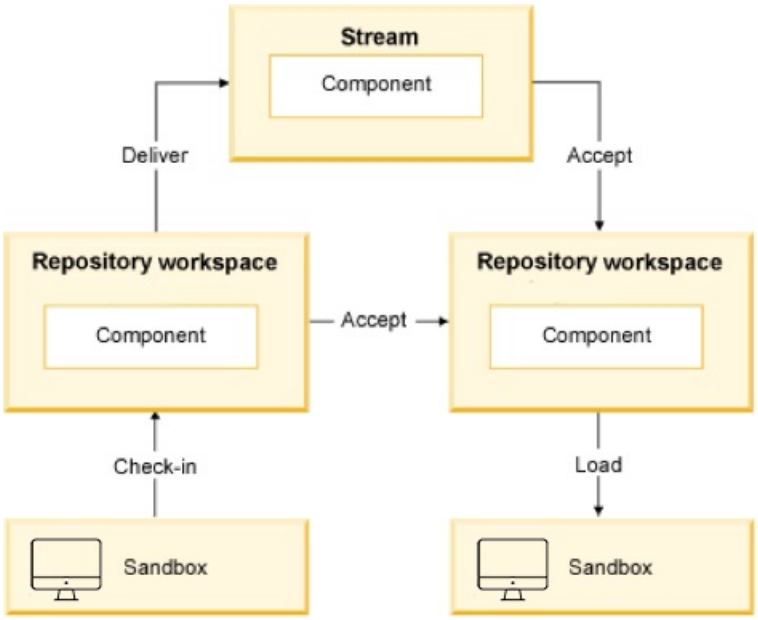


Source Management on RTC

RTC 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 RTC objects that are part of this process:

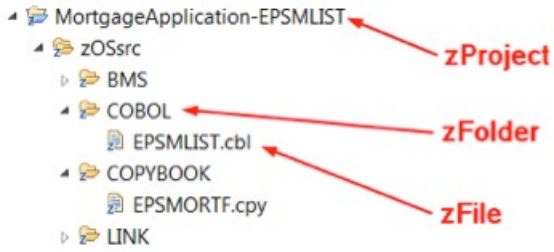
- **Stream** - represents a specific version, branch, or release of a collection of software components.
- **Component** - represents a collection of related software objects, such as a COBOL source library and COBOL copybook library. Together can be used to build a software component.
- **Repository Workspace** - represents a container object in RTC 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 RTC Objects

Special versions of RTC source control objects that apply specifically to z/OS:

- **zProject** - represents a type of RTC Eclipse project that contains a z/OS software project.
- **zComponent** - represents a specialized type of RTC component that contains zFolders.
- **zFolder** - is an RTC 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

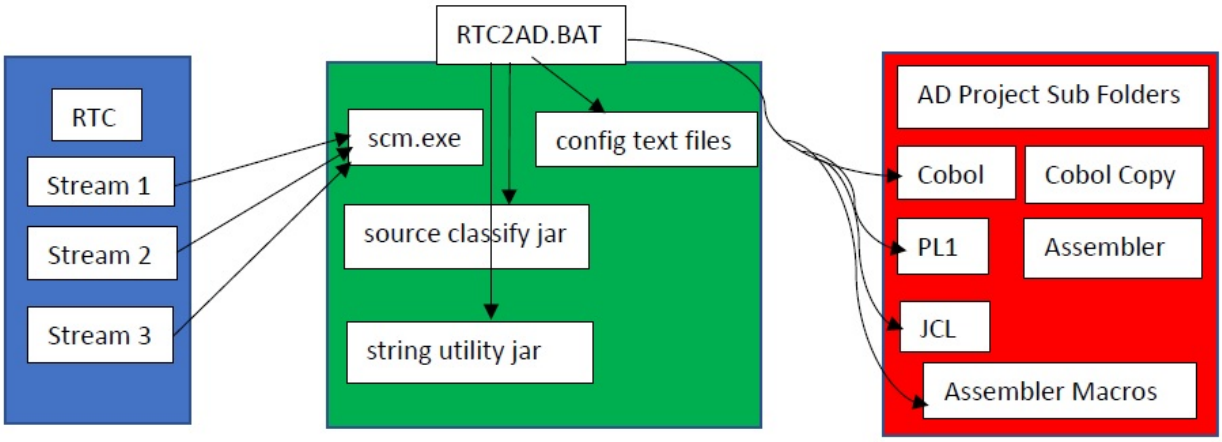
The RTC2AD integration tool is a Windows™ script, **rtc2ad.bat**, that makes calls to several functions provided by the RTC CLI program **scm.exe**. The IBM tool is distributed with the RTC Eclipse client (and possibly with other products). The assumption is that every RTC user has access to this tool.

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 RTC jazz server. It decides which RTC 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

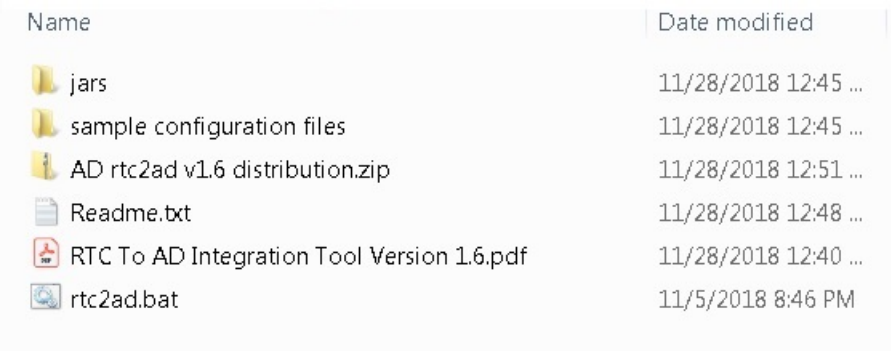
About this task

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:

Procedure

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 master configuration file, originally named `config.txt` in the distribution. For more information, see "Configuration File Keyword Refence" on page 31 or refer to the comments in the master 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 31 or refer to the comments in the master configuration file.
5. Copy both JAR files in the `jars` folder to an appropriate location. Ensure that RTC2AD_CLASSPATH in the master config file points to this folder.

Running the RTC2AD Tool

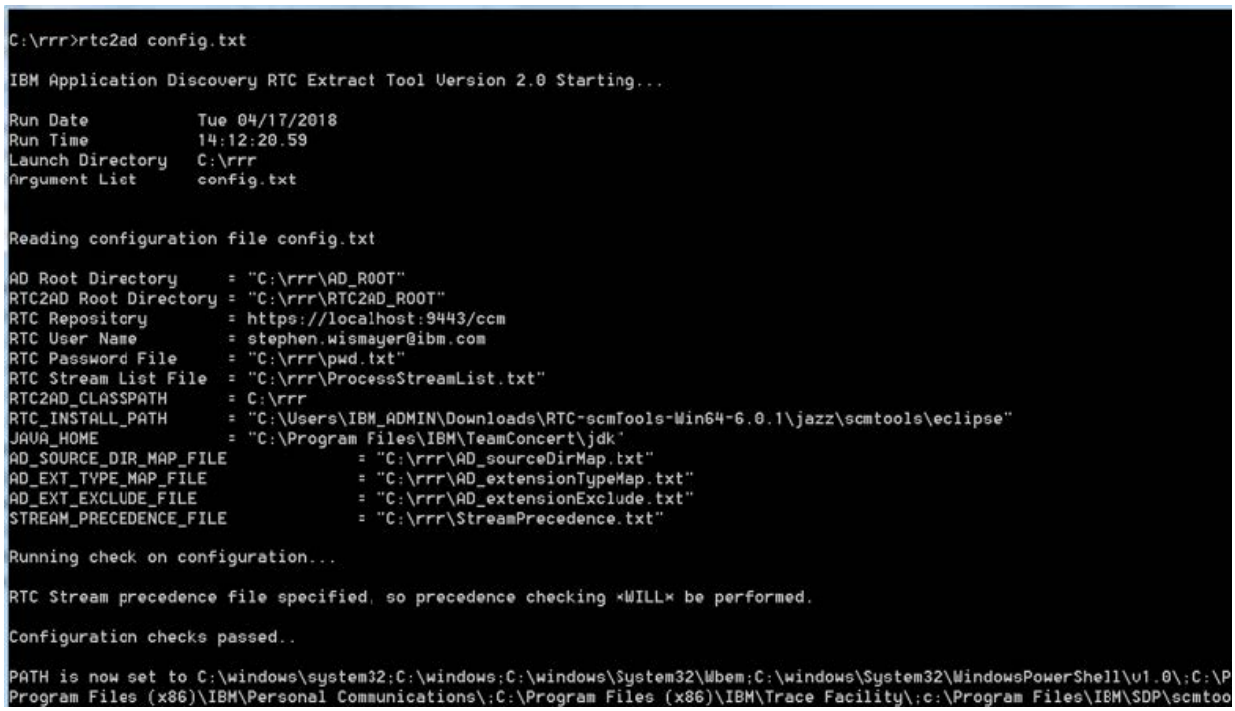
About this task

Follow these steps to run the RTC2AD tool:

Procedure

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 RTC 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 RTC change sets within a specified date interval are retrieved and classified. It greatly reduces the number of RTC 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 31.

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 RTC. The expected source types are designated in the AD_SOURCE_DIR_MAP_FILE.
RTC2AD_ROOT_DIR	Represents the root directory that serves as RTC sandbox. The sandbox is a set of directories on the local file system where RTC files and metadata are saved by the RTC 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 RTC repository where the zProject files are stored. It is the same URL that is used when a repository connection is created in the RTC Eclipse Client.
RTC_USERID	Represents the user ID under which the tool will log on to the RTC repository server. This user ID must have the appropriate access rights to the RTC 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 RTC 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 RTC streams that the tool needs to process. By default, the tool processes automatically every RTC 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 RTC CLI **scm.exe**. It is made available as part of the RTC Eclipse installation. It is required by `rtc2ad.bat`.
Java™_HOME	Represents the directory that contains the Java™ SDK or JRE installation.

Keywords	Description
RTC_COMP_LIST_FILE	Represents an extra file that can be used to restrict which RTC components are processed within the streams and specified in RTC_STREAM_LIST_FILE. Only RTC 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.
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 RTC 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 RTC 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 RTC 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 RTC 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 RTC 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.

Keywords	Description
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 RTC server and download the specified source files into the local RTC sandbox. The N value indicates that no connection needs to be made to the RTC server. The processing needs to be performed by using only the files already stored in the local RTC sandbox. The option is useful only in testing, saving time by not repeating downloads from the RTC server.
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 RTC_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 RTC 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 RTC workspace and load it with the RTC stream.
    b. Retrieve the list of all components in the stream from RTC, save in temporary working file.
    c. Follow these steps for each component in the stream:
        1) If `pass list` file exists, check whether the component is on it, else skip component.
        2) If no `pass list` file exists, process every component in the stream.
        3) Retrieve the list of all objects contained in the RTC component.
        4) Filter out only the zFile names and save them in a local file.
        5) Load all the zFiles from RTC into the local file system sandbox.
        6) 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.

7) After each file in the component is processed, build a file of copy commands.

8) A copy command is generated only if the file source type is resolved and type maps to AD folder.

9) Run the batch of copy commands, copying the file from local sandbox to the correct AD sub folder.

10) Repeat the loop to process the next RTC component defined in the stream.

8. When all RTC components in stream are processed, loop to the next stream specified in config.
9. When all specified streams are processed, delete the temporary RTC workspace.
10. Log off from RTC Jazz Team Server.

Chapter 7. Error Codes

Communication Errors

Error Code	Description
01	Invalid *lu* name or *lu* does not exist. **Description**: The LU name provided in the application entry is not defined in the external environment. **Action**: Correct the application entry to correspond with a valid partner LU alias or system ID. Refer to your support personnel in order to find the valid system identifications.
02	No free session is available. **Description**: When starting a conversation Comm/Booster finds that no sessions are available in order to start a conversation. This could happen because of a condition that tasks are hanging on the connection. **Action**: If tasks are hanging, free the connection by deactivating it or by deleting the tasks that are hanging. Try to determine the cause of hanging tasks (memory, deadlocks etc.). If this is not caused by hanging tasks, check the possibility of increasing the number of sessions.
03	Terminal options error. **Description**: Invalid connection definition, the connection was probably defined incorrectly. **Action**: Contact support personnel in order to check the connection.
04	Invalid request -internal error. **Description**: Comm/Booster has issued an invalid APPC request. This might be an internal logic failure. **Action**: Contact technical support.
05	Session not bound, allocation failed. **Description**: Conversation cannot be started because the session is not bound. This could be a configuration problem or an operational problem. **Action**: If this connection has worked before, contact the network operators in order to activate the connection.
06	
07	Lost connection. **Description**: The connection was lost during the conversation. The most probable cause for this is that the remote application has abended. If this is not the case, the session was deactivated by force. **Action**: Find the cause of the remote application bend and correct the problem. If the remote application was not abended check whether the connection was force deactivated.

Environment Errors

Error Code	Description
11	Invalid environment request-internal error. **Description**: One of the environment modules of Comm/Booster has issued an invalid request. **Action**: Contact technical support.
12	File is disabled. **Description**: Comm/Booster has tried to access a disabled file. **Action**: Enable all the files of Comm/Booster before attempting to use Comm/Booster.
13	File not open. **Description**: Comm/Booster has tried to access a closed file. **Action**: Open all the files of Comm/Booster before attempting to use Comm/Booster.
14	I/O error on file. **Description**: An I/O request has failed to one of the files of Comm/Booster. **Action**: Check that the application table file and the log file (if present) are defined correctly and accessible, correct the program and try again.
15	Record not found. **Description**: A record in one of the files was not found. This is probably an internal logic error. **Action**: Contact technical support.
16	Security violation. **Description**: The Service platform has rejected the user id and password of this transaction. If the service platform contains a security exit, this exit has rejected the security information. Otherwise, the security features of the service platform have rejected the conversation on security grounds. **Action**: Correct the user id and password and re-run the transaction. If the service platform contains a security exit, check that the exit is justified in rejecting the security information. If the service platform supports APPC security (i.e.CICS-RACF) check the security rules to correct the problem. If the user id and password do not match the client program should contain logic to notify the user that he is not logged on correctly.
17	Data set does not exist. **Description**: The application table or log file has not been found. **Action**: Check whether the product installation has been successful completed.
18	Invalid storage length. **Description**: An internal logic problem has occurred. This might happen as a result of a storage violation. **Action**: Check whether your program has corrupted storage.
19	Storage not available for **getmain**. **Description**: Comm/Booster has attempted to acquire storage but has failed. Check that at least 100K of storage is available for each transaction. **Action**: Correct the storage shortage and re-run the transaction.
20	Transaction id error.

Error Code	Description
	Description: Comm/Booster has attempted to start one of its internal transactions but the transaction id does not exist (mainframe only). **Action**: Check whether the product installation has been successful completed.
21	End of data on retrieve. **Description**: An internal transaction has abended because no parameters were available to it. This is an internal error. **Action**: Contact technical support.
22	Length error. **Description**: An internal length error has been detected. **Action**: Contact technical support.
23	Program id error. **Description**: Comm/Booster has attempted to start one of its internal programs but the program id does not exist. **Action**: Check whether the product installation has been successful completed.
24	No disk space. **Description**: There is not enough disk space on one of the files. **Action**: Make sure that enough disk space is available for the applications table and log file.
25	Duplicate error. **Description**: This is an internal logic error. **Action**: Contact technical support.
26	End of file. **Description**: This is an internal logic error. **Action**: Contact technical support.
27	ENQ error. **Description**: This is an internal logic error. **Action**: Contact technical support.

Logging Errors

Error Code	Description
30	Invalid logging state. **Description**: This is an internal logic error. **Action**: Contact technical support.
31	Invalid function. **Description**: This is an internal logic error. **Action**: Contact technical support.

Error Code	Description
32	Record not found. **Description**: This is an internal logic error. **Action**: Contact technical support.
33	Invalid function on slave side. **Description**: This is an internal logic error. **Action**: Contact technical support.
34	
35	End of chain. **Description**: This is an internal logic error. **Action**: Contact technical support.

Conversation Errors

Error Code	Description
42	Invalid request on first call. **Description**: A client or service program has called Comm/Booster with a storage field equal to nulls (beginning of new conversation), but the application request field did not contain a valid request for a conversation initiation. This is most probably a programming error. **Action**: Correct the program and re-run the conversation. Refer to chapter 2 for conversation rules.
43	Error on getting application record. **Description**: The application record of the requested service application is not found at conversation initialization, and default entry is not defined. **Action**: Check the application table to check whether the application exists or whether a default entry is present.
44	Server was called after session termination. **Description**: A Comm/Booster request was made with an invalid conversation ID. **Action**: Correct the program according to the specified protocol.
45	Actual length greater than buffer length. **Description**: The value of actual length in the connection record is greater than the value of buffer length. This is invalid. **Action**: Correct the program to enter a consistent size.
46	Negative buffer size. **Description**: The buffer size in the connection record has been initiated to a negative value. **Action**: Correct the program to enter a valid buffer size.
47	Buffer size is over 32500. **Description**: The buffer size in the connection record has been initiated to a value above 32500 which is the maximum buffer size.

Error Code	Description
	Action: Correct the program to enter a valid buffer size.
48	Conversation state error. **Description**: A conversation state error has been encountered. The application has issued a request incompatible with the current state. This is probably due to a programming error. **Action**: Correct the program according to the specified protocol.
49	Quit received from partner. **Description**: The partner application has issued a quit request. The conversation has ended and a backout has been performed by Comm/Booster for transactions using level 2 recovery. **Action**: If this situation is acceptable in the protocol, the application should handle this as a remote quit request.
50	On recovery - record for recovery found in partner log file. **Description**: This is an internal logic error. **Action**: Contact technical support.
51	Bad storage field. **Description**: This is an internal logic error. **Action**: Contact technical support.
52	Invalid version. **Description**: The version field in the connection record is invalid. Comm/Booster cannot determine the format of the connection record. **Action**: Correct the program according to the specified protocol.
53	Storage pointer is bad. **Description**: The storage field in the connection record is invalid. Comm/Booster cannot find the specified storage. The storage field has probably been overwritten. **Action**: Correct the program to handle the storage field according to protocol.
54	Data received is larger than expected. **Description**: The data received from the remote application is larger than the value in the buffer length field. The data has been returned to the application but was truncated. In this case the conversation is not aborted. **Action**: Correct the program to specify a correct buffer length field.
55	Security violation, execution denied. **Description**: Invalid security information has been provided. **Action**: Correct the program to specify the correct security information.
56	On recovery session - complete with controlled exit. **Description**: Recovery was initiated and completed successfully. **Action**: Exit the session.

Workstation Specific Errors

Error Code	Description
70	APPC interface error. **Description**: The APPC interface in the workstation has encountered an error. The APPC interface has probably abended. **Action**: Determine what the problem with the APPC interface is. If the problem reoccurs, contact the vendor of the APPC subsystem.
71	LU session limit exceeded. **Description**: The number of different APPC sessions has exceeded those allowed by the APPC interface. This is a workstation configuration problem. **Action**: Reduce the number of APPC sessions.
72	Too Many Conversations. **Description**: The number of ongoing conversations has exceeded the defined maximum that is allowed between the two platforms. **Action**: There may be a need to configure more sessions between the two platforms. If the problem recurs, some transactions may be hanging on the connection. If this is the case, determine the cause of the problem and correct.
73	Internal Error. **Description**: This is an internal logic error. **Action**: Contact technical support.
74	Internal Error. **Description**: This is an internal logic error. **Action**: Contact technical support.
75	Internal Error. **Description**: This is an internal logic error. **Action**: Contact technical support.
76	Reserved LU is in use. **Description**: The specified logical unit name has been reserved, and in use. **Action**: Do not use a reserved LU name.
81	Async: connection record not yet arrived. **Description**: A connection record has been passed to the non-blocking interface but the result is not yet available. This happens when the non-blocked application tries to check whether a reply is available yet. This is not necessarily an error and should be handled by the application. The conversation continues. **Action**: The calling program must retry later with the same connection record.
82	Async: Timeout. **Description**: A timeout has occurred when communicating through the non-blocking interface. This is similar to error type 81. **Action**: The calling program must retry later with the same connection record.
83	SRVRxT time out.

Error Code	Description
	Description: A call to SRVRQT has timed out. The problem is probably caused by the remote application which is not responding. The session is cleaned up and the sockets are closed. **Action**: Correct the remote application or supply a larger time out value.
84	User canceled conversation “simulated” timeout. **Description**: The user previously defined a simulated timeout which has now been emulated. **Action**: Further processing can be determined by the user.
90	Applrec not found in application table. **Description**: The application ID refers to an application table record not found in the application table. **Action**: Review the application table and correct as necessary.
91	Application table missing or Invalid. **Description**: The application table is missing in the path specified by the SRVAPPL specification and in the working directory. **Action**: Specify the SRVAPPL environment variable to point to a valid application table or create a valid application table in your working directory. For more information see chapter 5.
92	Workstation communication error. **Description**: The connection was lost during the conversation. The most probable cause for this is that the remote application has abended. If this is not the case, the session was deactivated by force. **Action**: Find the cause of the remote application abend and correct the problem. If the remote application was not abended check whether the connection was force deactivated.
93	Data conversion failed. **Description**: The data received or sent includes characters that cannot be converted from ASCII to EBCDIC or vice versa. **Action**: If the data is binary, correct the program to indicate binary data. Otherwise change the program to send only printable characters.
94	Conversion of data failed - malformed ASN1 data. **Description**: The data received or sent includes characters that cannot be converted from ASCII to EBCDIC or vice versa. **Action**: If the data is binary, correct the program to indicate binary data. Otherwise change the program to send only printable characters.
95	Data (de-) compression failed. **Description**: This is an internal logic error. **Action**: Contact technical support.
96	Timeout maximum entries reached. **Description**: The user has initiated a number of timeout requests in excess of the storage limits defined by Comm/Booster. **Action**: Contact technical support.

Documentation Notices for IBM Application Discovery for IBM Z

This edition applies to version 5.1.0 of IBM Application Discovery for IBM Z with the corresponding fix packs.

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 might 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 might 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 that are 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 can 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. Therefore, IBM 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 Copyright and trademark information.

GC27-8974-06