

IBM Tivoli Asset Discovery for z/OS
Version 8 Release 1

*Planning, Installation, Deployment, and
Migration Guide*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 33.

Contents

Chapter 1. Planning for deployment. 1

Predeployment considerations	1
Deployment data processes	1
Deployment for a single Repository.	2
Deployment for multiple Repositories	3

Chapter 2. Installing and customizing IBM Tivoli Asset Discovery for z/OS. 5

Installation prerequisites	5
Security and authorization prerequisites	5
Checklist of installation and customization tasks	7
Installing target libraries	9
Preparing DB2 database prerequisites.	10
Preparing local environment settings	11
Configuring a test Repository database	15
Creating a test Repository database in DB2	15
Creating a test Repository database in SQLite	15
Populating the test Repository database with data	15
Configuring a production Repository database	17
Creating a production Repository database	17
Configuring security for the production Repository database	18
Automating data collection and reporting activities	18
Maintaining the production Repository database	19

Chapter 3. Implementing deployment scenarios 21

Scenario 1: Implementing a single Repository database with a single GKB database	21
Scenario 2: Implementing multiple Repositories with a shared GKB database	22
Scenario 3: Implementing multiple Repositories with multiple GKB databases	23
Scenario 4: Collecting and transferring Inquisitor and usage data from remote sites	25
Scenario 5: Implementing in a sysplex environment	25

Chapter 4. Migrating to IBM Tivoli Asset Discovery for z/OS, version 8.1 27

Migrating to Tivoli Asset Discovery for z/OS from an earlier version	27
Migrating from version 7.5 to Tivoli Asset Discovery for z/OS version 8.1 (DB2 database)	27
Migrating from version 7.5 to Tivoli Asset Discovery for z/OS version 8.1 (SQLite database)	28
Migrating from version 7.2 to Tivoli Asset Discovery for z/OS version 8.1 (DB2 database)	29
Migrating from version 7.2 to Tivoli Asset Discovery for z/OS version 8.1 (SQLite database)	30
Migrating from Tivoli License Compliance Manager for z/OS, version 4.2 to Tivoli Asset Discovery for z/OS, version 8.1	31

Notices 33

Trademarks	34
----------------------	----

Chapter 1. Planning for deployment

Before you deploy IBM® Tivoli® Asset Discovery for z/OS®, consider which deployment option is best suited to your environment.

Related tasks:

Chapter 3, “Implementing deployment scenarios,” on page 21

Most implementations of Tivoli Asset Discovery for z/OS are based on one of the common deployment scenarios. An example is provided for implementing each of these common deployment scenarios with a DB2® Repository database. You can adapt an example for use with a SQLite Repository database.

Predeployment considerations

You can deploy Tivoli Asset Discovery for z/OS to use a single Repository or multiple Repositories.

Implementing a deployment in a single Repository is relatively straightforward because the data from all systems is imported into a single Repository. Before you deploy with a single Repository, plan the following aspects of the deployment:

- How frequently will the Inquisitor scan each system?
- If you have systems that are located at remote sites, what mechanism will transfer collected Inquisitor and usage data via file transfer protocol (FTP) to the central site, and how frequently will these transfers occur?
- How often is it necessary to load Inquisitor and usage data from each system into the Repository?

To deploy with multiple Repositories, plan the following aspects of the deployment:

- How frequently will the Inquisitor scan each system?
- How many Repositories to include in the deployment and are all of these Repositories located at the central site?
- For each system, which Repository loads the Inquisitor and usage data for the system?
- If you have systems that are located at remote sites, what mechanism will transfer collected Inquisitor and usage data via file transfer protocol (FTP) to the central site, and how frequently will these transfers occur?
- How often is it necessary to load Inquisitor and usage data from each system into its specified Repository?

Deployment data processes

Tivoli Asset Discovery for z/OS is structured on several key data processes.

Inquisitor data

The Inquisitor scans DASD volumes for libraries containing load modules and HFS/zFS files for z/OS UNIX program objects and produces Inquisitor data. These load modules and program objects are matched and associated to a particular vendor and product and the matched information is then loaded into the Repository tables. These processes are performed by running the Inquisitor Import job.

Usage event

A usage event describes a unique load of a load module or program object for an address space that can contain an account code. The Usage Monitor records these usage events as they occur on a particular operating system. After the usage data is imported into the Repository, each usage event is identified by the load module name, library name, and volume. It can then be associated to a particular product discovered on that system.

Repository

The Repository is a collection of database tables that contain processed Inquisitor and Usage Monitor data. To ensure that accurate data is stored in the Repository tables, the following criteria must be met:

- The DASD VOLSERS of the data being imported must be unique unless the DASD VOLSERS are shared or are clones of each other with identical contents.
- The data imported must be from systems with unique SMF IDs.

When you are designing the scope of a Repository, there are a few common scenarios that most installations fit into. It is common to define the scope of a Repository based upon a data center. In this scenario, each data center in the organization has a separate Repository.

CAUTION:

Import only DASD volumes with a unique VOLSER into your Repository.

The only way to prevent this sharing from taking place is to divide the z/OS systems with conflicting DASD/SMF IDs into separate Repositories. This can entail running one Repository for each sysplex or stand-alone z/OS system. With Tivoli Asset Discovery for z/OS, it is common for IT service providers to define separate Repositories for each customer. This definition also satisfies the need for separation of data and ease of reporting.

It is recommended to have a central DB2 subsystem or SQLite databases that contain all the Repositories in your entire enterprise. The usage and Inquisitor data that require processing should be transmitted to this central DB2 subsystem or SQLite database by using the Tivoli Asset Discovery for z/OS Automation Server or equivalent automation product.

Deployment for a single Repository

The recommended procedure for deploying the Inquisitor and Usage Monitor to collect raw data is to deploy both components on every system in your organization.

After you deploy both components to each system in your organization, perform data collection in the following sequence:

1. Use the Inquisitor Job to scan all available DASD on each z/OS System.
2. Import Inquisitor data by running the Inquisitor Import job.
3. Ensure that the Usage Monitor is active on all z/OS systems, directly after IPL.
4. Import Usage data by running the Usage Import job. Run this job after Inquisitor data has been imported.

Tivoli Asset Discovery for z/OS displays products that have been discovered. Usage data collected from every system by the Usage Monitor is imported and usage events are assigned to the discovered products, enabling analysis of product use by system.

The first step in deploying Tivoli Asset Discovery for z/OS is to run the SMP/E installation of the product, followed by the customization and creation of the database resources.

The next step is to create a test Repository. This deployment exercise is useful as it helps you to:

- Gain familiarity with the product.
- Check that your Repositories are defined correctly in terms of your business requirements and that the DASD VOLSERs and SMF IDs are unique.
- Ensure that data-sizing is adequate.
- Analyze the integrity of the data.

As part of this test implementation, you can then deploy the Inquisitor and Usage Monitor to all systems in your organization. It is advisable to first start the Usage Monitor on every system, in order to gather a significant amount of usage data. Place the test repository on a test or development DB2 subsystem.

At this point you can start the Tivoli Asset Discovery for z/OS Analyzer and connect to the Repository. To verify the data collected by the Inquisitor and Usage Monitor, log on to the Analyzer and navigate to the Discovery menu tab. From this menu you can proceed to various reports on discovered products and module usage.

After you move your Repositories to their final location, you should consider setting up automation of the product.

Deployment for multiple Repositories

Multiple Repositories can be required to provide support for more than one data center, for different geographical regions, and for running multiple customers.

You can locate multiple Repositories in one central location, or you can locate them in geographically dispersed locations. Multiple Repositories may be organized as follows:

1. For a central location
2. For geographically dispersed locations

Central location

Each Repository contains data that is divided up into logical units, for example:

- Data center
- Outsourced customer
- Sysplex

Each Repository has its own database. For DB2, all repositories must reside in the same DB2 subsystem but for SQLite, each repository must reside in its own SQLite database. For DB2 only, the advantage of this configuration is that reporting can be performed on data across all repositories. With this configuration, all repositories

can share the same Global Knowledge Base (GKB) and you only have to maintain a single copy of the GKB.

Geographically dispersed locations

Each Repository is defined with its own database at a specific geographic site as a stand alone operation. Reporting can only be performed for each specific Repository. The disadvantage with this configuration is that it can be necessary to consolidate Repository data to a central site for reporting purposes.

Chapter 2. Installing and customizing IBM Tivoli Asset Discovery for z/OS

The product installation involves downloading the product and available updates, preparing the database, and configuring and populating a test database. After verifying that all components are correctly installed, you duplicate the test database to create a production database where you automate data collection and import tasks.

Installation prerequisites

Before you install IBM Tivoli Asset Discovery for z/OS, verify that the required hardware and software requirements are available in the installation environment.

Hardware requirements

The hardware requirements for running Tivoli Asset Discovery for z/OS are a z/Architecture[®] machine capable of running z/OS Version 1 Release 11 or later.

Software requirements

The software requirements for running Tivoli Asset Discovery for z/OS are:

- z/OS Version 1 Release 11 or later.
- Database can be either:
 - DB2, Version 9, Release 1 or DB2 Version 10, Release 1 if you choose DB2 for your Tivoli Asset Discovery for z/OS database
 - SQLite, Version 3.2.6.23.1, that is embedded in Tivoli Asset Discovery for z/OS

If you do not have a DB2 license, contact IBM support in order to install TADz with SQLite only. It is not necessary to install the database on all of your z/OS systems but it must be installed on at least one z/OS system

- Language Environment[®] for z/OS.
- Browser can be either:
 - Firefox ESR Version 10.0.10 with JavaScript and cookies enabled
 - Internet Explorer, Version 9 with JavaScript and cookies enabled
- Microsoft Excel 2003

Security and authorization prerequisites

A z/OS user ID is required with appropriate RACF[®] access to submit the batch jobs used in the customizing and operation of Tivoli Asset Discovery for z/OS. Additional security and authorization configurations can be necessary, depending on your environment.

RACF authorizations

The following table lists the RACF authority required to run Tivoli Asset Discovery for z/OS Started Tasks, Usage Monitor, Analyzer, and Automation Server. Consult with your RACF administrator to define the required RACF authority.

Table 1. RACF authority required for each started task

Started task name	SHSIMOD1	PARMLIB	SHSIANL1	SHSIANL2	ACDS	(DB2 only) SDSNLOAD and SDSNEXIT	HLQIDS data set	Usage Monitor output data sets
Usage Monitor	READ	READ	n/a	n/a	n/a	n/a	READ	ALTER
Analyzer	READ	READ	READ	READ	n/a	READ	n/a	n/a
Automation Server	READ	READ	n/a	n/a	CONTROL	n/a	n/a	n/a

The started task should be defined in the resource class STARTED, with additional detail in the STDATA segment of the resource. It can also be defined in the started task table ICHRIN03, but this requires an IPL to add or update a task definition.

For example:

```
RDEFINE STARTED HSI*.* UACC(NONE) +
STDATA (USER(uuuuuuu))
```

Replace *uuuuuuuu* with the name of the started task user for Tivoli Asset Discovery for z/OS

```
SETROPTS RACLIST(STARTED) REFRESH
```

For non-RACF security products, consult your Security Administrator.

z/OS UNIX security

Both the Usage Monitor and the z/OS UNIX Inquisitor need sufficient authority to navigate the UNIX file system. The writer task of the Usage Monitor requires access to resolve symbolic links, while the UNIX Inquisitor is tasked with discovering executable files.

APF

The Inquisitor and Usage Monitor use z/OS authorized system services. These programs are contained in the PDSE Load Library SHSIMOD1, which must be authorized using APF in order to run the Usage Monitor and/or the Inquisitor when the latter is not being run with PARM=NOAPF.

MAXCAD parameter

A z/OS system programmer must have the necessary authorities to perform this task.

The Usage Monitor uses a SCOPE=COMMON data space. For this reason, it is necessary to have at least two additional system-wide data space PASN entries. Tivoli Asset Discovery for z/OS uses one data space, and after a switch, creates a new one. The older data space is not deleted until it is processed by the Usage Monitor writer task.

To enable the creation of the Usage Monitor data spaces, increase the Usage Monitor MAXCAD system parameter by an additional value of 3 (three). For example, increase an existing installation with MAXCAD=100 to MAXCAD=103 to cater for the addition of TADz Usage Monitor data spaces. Define the MAXCAD parameter in the IEASYSxx member of the system PARMLIB library. For more information about the default and valid value range for this parameter, refer to the *MVS Initialization and Tuning Reference, SA22-7592*.

DB2 authorization

You need DB2 privileges to perform the following tasks:

- DBADM authority to access the product database. You may need to drop and create DB2 resources.
- BIND plans and packages.
- EXECUTE authority to execute plans and packages.
- SELECT authority to access the DB2 Catalog tables.
- LOAD, REPAIR, and STATS privileges to run DB2 utilities LOAD, REPAIR, and RUNSTATS.
- GRANT USE OF BUFFERPOOL privilege to use specific buffer pools.
- GRANT USE of STOGROUP privilege to use a specific storage group.
- Access to work file database or TEMP database for Declared Global Temporary table.

SQLite authorization

To perform an installation with a SQLite database requires that authority to perform the following tasks:

- Allocate, format and mount a zFS file system.
- Grant access to z/OS OMVS groups

Checklist of installation and customization tasks

This checklist includes a set of procedures that include installing the product, creating a test database, populating data, and validating the test installation. When you complete all of these procedures you are ready to create a production environment for IBM Tivoli Asset Discovery for z/OS.

Table 2. Checklist of installation and customization tasks

Step	Description	Data sets and members
1	<p>Install target libraries.</p> <p>A z/OS system programmer performs this task.</p> <p>Installing target libraries</p>	<p>hsi= IBM Tivoli Asset Discovery for z/OS product prefix</p> <ul style="list-style-type: none"> • hsi.SHSIANL1 • hsi.SHSIANL2 • hsi.SHSIEXEC • hsi.SHSIGKB1 • hsi.SHSIMJPN • hsi.SHSIMOD1 • hsi.SHSIPARM • hsi.SHSIPROC • hsi.SHSISAMP
2	<p>Prepare database prerequisites.</p> <p>A DB2 database administrator performs this task.</p> <p>“Preparing DB2 database prerequisites” on page 10</p> <p>The SQLite database is embedded in Tivoli Asset Discovery for z/OS and the prerequisites are already configured. If you plan to use the SQLite database for your implementation, you do not have to perform this task.</p>	<p>DB2 SDSNSAMP data set members:</p> <ul style="list-style-type: none"> • DSNTIJTM • DSNTIJCL

Table 2. Checklist of installation and customization tasks (continued)

Step	Description	Data sets and members
3	<p>Prepare local environment settings.</p> <p>Tasks include editing the HSISCUST member in the SHSISAMP target library, changing the SYSIN DD entry for local settings, and running the HSISCUST job. This job generates JCL jobs that you run in subsequent tasks.</p> <p>A database administrator and a Tivoli Asset Discovery for z/OS administrator perform this task.</p> <p>Preparing local environment settings</p>	<p>hsiinst=hlq for JCLLIB, and PARMLIB libraries</p> <p>hsi.SHSISAMP data set member: HSISCUST generates hsiinst.&DB.JCLLIB hsiinst.&DB.PARMLIB</p>
4	<p>Create a test Repository database.</p> <p>A database administrator and a Tivoli Asset Discovery for z/OS administrator perform this task.</p> <ul style="list-style-type: none"> • “Creating a test Repository database in DB2” on page 15 • “Creating a test Repository database in SQLite” on page 15 	<p>JCLLIB data set member:</p> <ul style="list-style-type: none"> • HSISDB01 • HSISDB02 • HSISDB03 • HSISGKBL • HSISGRNT
5	<p>Collect data and import it into the test Repository database.</p> <p>A Tivoli Asset Discovery for z/OS administrator performs this task.</p> <p>“Populating the test Repository database with data” on page 15</p>	<p>JCLLIB data set members:</p> <ul style="list-style-type: none"> • HSISINQZ: Gather Inquisitor data. • HSISINQU: Gather Inquisitor UNIX data. • HSISUMON: Gather Usage Monitor data. • HSISIQIM: Import Inquisitor data. • HSISUIMP: Import usage data.

Table 2. Checklist of installation and customization tasks (continued)

Step	Description	Data sets and members
6	<p>Create the production Repository database and arrange for regular maintenance.</p> <ul style="list-style-type: none"> • “Creating a production Repository database” on page 17 • “Maintaining the production Repository database” on page 19 	<p>JCLLIB data set members:</p> <ul style="list-style-type: none"> • HSISCUST • HSISDB01 • HSISDB02 • HSISDB03 • HSISGKBL • HSISGRNT • HSISGRTB • HSIJMON • HSIASALC • HSIJAUTO • HSIJANLO • HSISANS1 • HSISANS2 • HSISANS3 • HSISINQZ • HSISINQU • HSISUMON • HSISIQIM • HSISUIMP • HSISUDEL • HSISUSUM • HSISLDEL • HSISTPRM • HSISUN81 • HSISLO81 • HSISUT01 • HSISUT02 • HSISUT03 • HSISUT04 <p>PARMLIB data set member:</p> <ul style="list-style-type: none"> • HSISMNPM • HSIAPARM • HSISANP1

Installing target libraries

Before you can install Tivoli Asset Discovery for z/OS in a production environment, you can create a test environment.

Before you begin

The installation must be performed by a z/OS system programmer that has access to ShopzSeries to download the product.

Procedure

1. Download IBM Tivoli Asset Discovery for z/OS, Version 8.1, and all available maintenance components from ShopzSeries.
2. Follow the Receive and Apply instructions in the *Tivoli Asset Discovery for z/OS Program Directory* to install the target libraries. The following libraries are installed:

Data set low level qualifier (LLQ)	Description
SHSIANL1	Analyzer reports for Tivoli Asset Discovery for z/OS.
SHSIANL2	Java script for Tivoli Asset Discovery for z/OS.
SHSIEXEC	REXX code for Tivoli Asset Discovery for z/OS.
SHSIGKB1	Global Knowledge base data for Tivoli Asset Discovery for z/OS.
SHSIMJPN	Message templates in Japanese.
SHSIMOD1	Load modules for Tivoli Asset Discovery for z/OS.
SHSIPARM	Templates that the HSISCUST job uses to populate &HSIINST..PARMLIB library.
SHSIPROC	JCL PROCs for Tivoli Asset Discovery for z/OS.
SHSISAMP	Templates that the HSISCUST job use to populate the &HSIINST..JCLLIB library.

3. Install all PTF maintenance packages available on the Preventive Service Planning website.
4. Ensure that the target libraries are available to the LPAR where you intend to configure the test DB2 for z/OS database.
5. Specify that the SHSIMOD1 data set is authorized by the Authorized Program Facility (APF). For example, you can enter the following command:
SETPROG APF,ADD,DSN=hsi.SHSIMOD1,SMS
or
SETPROG APF,ADD,DSN=hsi.SHSIMOD1,VOL=xxxxxx
6. Schedule a change request to roll out target libraries to all z/OS LPARs where IBM Tivoli Asset Discovery for z/OS is used and include APF authorization for SHSIMOD1. For example, update the appropriate PROGxx member.

Preparing DB2 database prerequisites

The DB2 environment for the test z/OS installation includes various prerequisites that you must configure.

Before you begin

DB2 database administrator and Tivoli Asset Discovery for z/OS administrator privileges are required to perform this task.

DB2 for z/OS, Version 9 or Version 10, must be installed. DB2 must have access to a minimum of 1600 cylinders of 3390 DASD space.

Procedure

1. Run the DSNTIJCL job from DB2 SDSNSAMP to bind the DSNACLI plan and enable the Call Library Interface (CLI/ODBC) DB2 plan. If you encounter a SQL error, code 805, rebind this plan with the latest DB2 maintenance package and include the following package in the job:
BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIMS) - CURRENTDATA(YES)
ENCODING(EBCDIC) SQLERROR(CONTINUE)
2. Run the DSNTIJTM job from DB2 SDSNSAMP to bind the DSNREXX plan and enable the REXX DB2 plan.

Preparing local environment settings

After installation, you can create a custom version of any job in the JCLLIB library or any parameter in the PARMLIB library, by copying and editing the relevant job in the HSISCUST member in the hsi.SHSISAMP data set.

Depending on your environment, you can define parameters for the following environments:

- DB2
- SQLite
- Remote configuration

The **DBTYPE** parameter determines the environment and creates the jobs to customize and run the product in that environment.

Review the HSISCUST job parameters before you begin. A database administrator and a system programmer are required to perform the customization. After you make the required changes, submit the job. The JCL creates or reuses two output PDSE libraries and two sequential data sets.

The job creates the following PDSE libraries:

- The JCLLIB library contains a Job Control Language (JCL) script that implements and operates the product.
- The PARMLIB library contains predefined parameters that the JCL script references.

The sequential data sets are:

- The UM.HLQIDS sequential data set is referenced by the Usage Monitor on creation, and contains a single record.
- The TADZLOCK sequential data set is a dummy file used for serialization.

General parameters

The following table lists the general parameters that you must consider for all environments.

Table 3. General customization parameters

Parameter	Description
SET HSI	You must set this JCL parameter to the high-level qualifiers of the target libraries created by the SMP/E installation process. The default parameter is HSI.V810.

Table 3. General customization parameters (continued)

Parameter	Description
SET ISP	The customization tool uses ISPF services to customize the parameters and JCL for the user. This parameter specifies the high-level qualifiers for the ISPF target libraries. The default parameter begins with ISP.
DBTYPE	This parameter determines the environment and creates the JCL and parameters for that environment: <ul style="list-style-type: none"> • DB2 • SQLITE • REMOTE: The product collects Inquisitor and Usage Monitor data at remote sites and no database is required.

Required settings for all database types

The following table lists the required settings for all databases.

Table 4. Required settings for all databases

Parameter	Description
CLASS	CLASS
MSGCLASS	JES message class
MSGLEVEL	JES message level.
CEERUN	This parameter specifies the fully qualified Language Environment CEERUN data set.
CBCDLL	This parameter specifies the fully qualified Language Environment CBCDLL C++ runtime data set.
HSINST	This parameter specifies the high-level qualifiers of the JCLLIB and PARMLIB data sets that are created by running the HSISCUST job. If the JCLLIB and PARMLIB data sets exist, they are reused and you can replace members with updated information. Two other sequential data sets are either created or reused. The name specified for this parameter must be less than, or equal to, 19 characters in length.

Settings for DB2 and SQLite databases

The following table lists the settings for DB2 and SQLite databases.

Table 5. Settings for DB2 and SQLite databases

Parameter	Description
SYS	System where database resides
REPZSCHM	This parameter is used as a full qualifier for the tables and index definitions in the repository, and as a part qualifier for the tables and index definitions in the local knowledge base, and local knowledge base for z/OS UNIX. The REPZSCHM name must be less than, or equal to, 8 characters in length. If you are migrating from Tivoli Asset Discovery for z/OS, Version 7.5 to Version 8.1, the value specified for this parameter must be the same as defined for the DB parameter. If you specify a different value, the migration will fail.

Table 5. Settings for DB2 and SQLite databases (continued)

Parameter	Description
GKBZSCHM	<p>This parameter is part of the table qualifier and the index definitions qualifier for the GKB, GKB for z/OS UNIX, and Inquisitor filters. The GKBZSCHM name must be less than, or equal to, 8 characters in length.</p> <p>If you are migrating from Tivoli Asset Discovery for z/OS, Version 7.5 to Version 8.1, the value specified for this parameter must be the same as defined for the DBGKB parameter. If you specify a different value, the migration will fail.</p>
DBADMIN	<p>DBADMIN is an optional parameter. For a DB2 database, this parameter specifies the list of user IDs that are granted administrator access to the database and its contents. Specify an empty string if you do not want to grant administrator access to user IDs for the database specified in DB and DBGKB. For SQLite, this parameter specifies the list of user IDs that can connect to the z/OS RACF group.</p>
SIZE	<p>This parameter specifies the initial space allocations for DB2 and SQLite table spaces of the three largest tables. The default value of SIZE is 1.</p>

DB2 database settings

The following table lists the DB2 database settings.

Table 6. DB2 database settings

Parameter	Description
DB	<p>This parameter specifies the name of the repository database that the product uses to store all of the information that it gathers other than from the GKB. The DB name must be less than, or equal to, 8 characters in length.</p>
DBGKB	<p>This parameter defines a single GKB database that is accessed by multiple repositories under the same DB2 subsystem. The DBGKB name must be less than, or equal to, 8 characters in length, and must not have the same name as the name defined for the DB.</p>
DB2LOAD	<p>This parameter specifies the fully qualified SDSNLOAD data set name.</p>
DB2EXIT	<p>This parameter specifies the fully qualified SDSNEXIT data set name. If the DB2EXIT library does not exist, use the same value as the DB2LOAD parameter.</p>
DBSSID	<p>This parameter specifies the DB2 subsystem ID on the z/OS System.</p>
LOC	<p>This parameter specifies the CLI/ODBC location for the DB2 subsystem ID on the z/OS system. You can use the DB2 DISPLAY DDF command to determine the Location.</p>
SETSQLID	<p>This parameter is used in SET CURRENT SQLID to allow a different user to define DB2 objects. This parameter is optional. The SETSQLID value must be less than, or equal to, 8 characters in length.</p>

Table 6. DB2 database settings (continued)

Parameter	Description
SGHSITAB	This parameter specifies the storage group name for small tables in the database. The default value is SGHSITAB (same as the parameter name). Consult your DB2 database administrator for security implications and naming conventions. See the SQL statement CREATE STOGROUP for more information.
SGHSIBIG	This parameter specifies the storage group name for large tables in the database. The default value is SGHSIBIG (same as the parameter name). Consult your DB2 database administrator for security implications and naming conventions. See the SQL statement CREATE STOGROUP for more information.
SGHSIIDX	This parameter specifies the storage group name for indexes in the database. The default value is SGHSIIDX (same as the parameter name). Consult your DB2 database administrator for security implications and naming conventions. See the SQL statement CREATE STOGROUP for more information.
SGTABCAT	This parameter specifies the VCAT of the DB2 table space data set names for small tables in the database. Consult your DB2 database administrator for security implications and disk storage requirements. This parameter is referenced by storage group name parameter SGHSITAB.
SGTABVOL	This parameter specifies the names of the volumes that the table space data sets for small tables are allocated on. This parameter is referenced by storage group name parameter SGHSITAB.
SGBIGCAT	This parameter specifies the VCAT of the DB2 table space data set names for large tables in the database. Consult your DB2 database administrator for security implications and disk storage requirements. This parameter is referenced by storage group name parameter SGHSIBIG.
SGBIGVOL	This parameter specifies the names of the volumes that the table space data sets for large tables are allocated on. This parameter is referenced by storage group name parameter SGHSIBIG.
SGIDXCAT	This parameter specifies the VCAT of the DB2 data set names for indexes in the database. Consult your DB2 database administrator for security implications and disk storage requirements. This parameter is referenced by storage group name parameter SGHSIIDX.
SGIDXVOL	This parameter specifies the names of the volumes that the data sets, for indexes, are allocated on. This parameter is referenced by storage group name parameter SGHSIIDX.
<ul style="list-style-type: none"> • BPDB • BPTS • BPIX 	These parameters specify the buffer pool definitions for the database, table spaces, and indexes. See Appendix D, "Performance and tuning," on page 241.

SQLite database settings

The following table lists SQLite database settings.

Table 7. SQLite database settings

Parameter	Description
SQLTZFS	zFS linear vsam dataset name that is used for Tivoli Asset Discovery for z/OS SQLite databases.

Table 7. SQLite database settings (continued)

Parameter	Description
SQLTPATH	USS directory where the SQLTZFS dataset is mounted. The HSISDB01 job in JCLLIB creates this path at a later time.

Configuring a test Repository database

Configuring the test Repository database includes setting up database objects, creating the Global Knowledge Base (GKB) database, and configuring access to the Repository and the GKB databases. Most sites maintain both a test Repository database and a production Repository database. After you configure and validate the test Repository database, repeat this task to create the production Repository database.

Creating a test Repository database in DB2

Creating the Tivoli Asset Discovery for z/OS database includes setting up storage groups, the database name, and the administrator logon details. You also create the Global Knowledge Base (GKB) environment and then grant access to the database.

Procedure

1. Run the HSISDB01 job to create the storage groups.
2. Run the HSISDB02 job to create the database objects for the GKB.
3. Run the HSISDB03 job to create the Repository database and database objects.
4. Run the HSISGKBL job to load the GKB.
5. Run the HSISGRNT job to grant DBADMIN access to the Tivoli Asset Discovery for z/OS administrator to the Repository and the GKB databases.

Creating a test Repository database in SQLite

Creating the test Repository database includes allocating, formatting and mounting a zFS file system and also grant access to z/OS OMVS groups. Creating the Tivoli Asset Discovery for z/OS database includes setting up storage groups, the database name, and the administrator logon details. You also create the Global Knowledge Base (GKB) environment and then grant access to the database.

Procedure

1. Run the HSISDB01 job to allocate, format, and mount a zFS file system.
2. Run the HSISDB02 job to create the database objects for the Global Knowledge Base (GKB).
3. Run the HSISDB03 job to create the Repository database and database objects.
4. Run the HSISGKBL job to load the GKB.
5. Run the HSISGRNT job to grant access to the z/OS OMVS groups that the Tivoli Asset Discovery for z/OS administrator is a member of.

Populating the test Repository database with data

You can populate the test Repository database in stages. Begin by collecting and importing Inquisitor and Usage Monitor data on the local LPAR, and then verify that this process is successful before collecting and importing data from other LPARs.

Collecting and importing data to the test Repository database:

After creating the databases and database objects, you are ready to collect Inquisitor and Usage Monitor data. You can then import the collected data into the test Repository database.

Procedure

1. Run the HSIINQZ job to scan the DASD for z/OS product modules and generate output to the DD HSIPZIP output file. For large sites, this operation can take up to an hour. You can perform steps 3 and 4 while the job is running.
2. Run the HSIINQU job to scan z/OS UNIX files and generate output to the DD HSIXZIP output file. For large sites, this operation can take up to an hour. You can perform steps 3 and 4 while the job is running.
3. To run the Usage Monitor to gather initial usage data, perform the following tasks:
 - a. Run the HSIISUMON job to start the Usage Monitor as a batch job. The Usage Monitor is typically run as a started task, but you can run it as a batch job for this test. This job runs continually until you stop it manually and most of the time this job is idle.
 - b. Stop the Usage Monitor to generate the `hsiinst.UM&SMF.D*.T*` output file. For example, enter the following command to stop the started task:

```
P HSIJMON
```
4. To import Inquisitor (IQ) data into the test Repository database, perform the following tasks:
 - a. Verify that the HSIINQZ and HSIINQU jobs that you started in steps 1 and 2 have completed. If the jobs are still running, wait until they are completed. The output logs from these jobs provide information on the number of records collected.
 - b. Run the HSIISQIM job to import the data from the HSIPZIP and HSIXZIP output files that were created by the HSIINQZ and HSIINQU jobs. For large sites, this job can take at least 2 hours to run the first time. Performance is 90 per cent faster on subsequent runs.
5. Run the HSIISUIMP job to import usage data from the `hsiinst.UM&SMF.D*.T*` file.

Verifying the results of the data import with the Analyzer:

After you complete the collection and import of Inquisitor and Usage Monitor data, use the Analyzer to verify that the import was successful.

Procedure

1. Review the HSIISANP1 PARMLIB library settings and modify if necessary. These settings specify the Tivoli Asset Discovery for z/OS administrator user id and password.
2. Run the HSIISANLO JCLLIB job on the test Repository database. This job, typically, runs continually but you can enter the `F HSIISANLO, STOP` command to stop it.
3. On your PC browser, logon to the Analyzer utility with the values specified in the HSIISANP1 PARMLIB library.
4. Review the Analyzer reports to confirm that all expected products have been identified. If a product is missing, perform the following tasks to identify the reason why a product is not included:
 - Check that the product is in the GKB and report any missing product to IBM support so that they can provide an updated GKB for the product.

- If the product exists in the GKB, check that the product is installed on the test z/OS. If the product is not installed on the test z/OS, run the Inquisitor utility on a system where the product is installed and then import that data into the test database.

Collecting and importing data from other systems:

After you verify that all components are correctly installed on the test Repository database, you can now discover and import Inquisitor and Usage Monitor data from other z/OS logical partitions (LPARs).

Procedure

1. Run the following jobs to collect Inquisitor and Usage data from other systems:
 - a. Run the HSIINQZ job to scan all other LPARs and generate output to the `hsiinst.HSIPZIP.Z&SMF` file.
 - b. Run the HSIINQU job and generate output to the `hsiinst.HSIUZIP.U&SMF` file.
 - c. Run the HSIUMON job to start the Usage Monitor as a batch job on the other LPARS.
2. Transfer collected data to the central site via file transfer protocol (FTP).
3. Run the following jobs to import Inquisitor and Usage data at the central site:
 - a. Run the HSIQIM job to import Inquisitor data from the `hsiinst.HSIPZIP.Z&SMF` and `hsiinst.HSIUZIP.U&SMF` files for each LPAR.
 - b. Run the HSIUIMP job to import Usage data from the `hsiinst.UM&SMF.D*.T*` file for each LPAR.

Configuring a production Repository database

Most implementations include a test Repository database and a production Repository database. Configuring a production Repository database involves creating the database and importing data, configuring security, and automating data collection activities

Creating a production Repository database

The production Repository database runs on a development logical partition (LPAR) and it is not necessary to run it on a business workload LPAR. You can duplicate the content of test Repository database to populate production Repository database without collecting and importing Inquisitor and Usage Monitor data again.

About this task

You can create the production Repository database on a DB2 or SQLite database. This procedure combines instructions for both database environments. Refer to the instructions for creating a test Repository database if you require database-specific instructions.

Procedure

1. Run the HSIIDB01 job.
 - For DB2, the job creates storage groups.
 - For SQLite, the job allocates the zFS file system.
2. Run the HSIIDB02 and HSIIDB03 jobs to create the Global Knowledge Base (GKB) and Repository databases and database objects.
3. Run the HSIIDB04 job to load the GKB.

4. Run the HSIAGRNT job.
For DB2, the job grants DBADMIN access to the Tivoli Asset Discovery for z/OS administrator for the Repository and GKB databases.
For SQLite, the job grants access to the z/OS OMVS groups.
5. Run the HSIAGRTB job.
For DB2, this job grants SELECT access to database tables.
6. To populate the production Repository database, repeat the procedure for collecting and importing data that you performed to populate the test Repository database.

What to do next

Configure security for the production Repository database.

Configuring security for the production Repository database

Resource Access Control Facility (RACF) security provides authentication, authorization, and auditing control for working with z/OS systems.

Procedure

1. Define a profile in the STARTED class to associate a user ID with the HSIJMON, HSIJAUTO, and HSIJANLO started tasks.
2. Specify that user IDs have the following access permissions:
 - a. READ access to hsi** data sets
 - b. ALTER access to hsiinst.** data sets

What to do next

Configure the automation of data collection activities on the production Repository database.

Automating data collection and reporting activities

When you configure the Usage Monitor, the Automation Server, and the Analyzer to run as started tasks, these data collection and reporting activities are automated.

Procedure

1. Configure the Usage Monitor utility to start automatically:
 - a. In the HSIISMNPM member of the PARMLIB data set, modify settings if necessary so that the **DSN(hsiinst.UM&SMF)** command generates hsiinst.UM&SMF.D*.T* data sets.
 - b. Schedule a change request to roll out the new HSIJMON started task on all z/OS LPARs.
 - c. Copy the HSIJMON started task from the JCLLIB library to the system PROCLIB data set.
 - d. Arrange for the HSIJMON started task to start early in the initial program load (IPL) cycle to ensure that all usage activity is recorded.
2. Configure the Automation Server utility to start automatically and to automate data collection and import tasks:
 - a. Schedule a change request to roll out a new HSIJAUTO started task on all z/OS LPARs.
 - b. Run the HSIASALC job to define the automation control Virtual Storage Access Method (VSAM) data set.

- c. Configure the HSIAPARM settings to perform the following tasks every weekend:
 - Remote hosts: Runs an Inquisitor scan job to collect data, runs the ZCAT to amalgamate usage data, and transfers collected data via file transfer protocol (FTP).
 - Database host: Runs an Inquisitor import job, runs a usage import job, and aggregates the data.
 - d. Optional: If necessary, run the HSIASSCT job to mark existing data sets as being already processed in the automation control data set.
 - e. Copy the HSIJAUTO started task from the JCLLIB library to the system PROCLIB data set.
 - f. Arrange for the HSIJAUTO to start automatically at any time in the IPL cycle.
3. Configure the Analyzer utility to start automatically:
 - a. Schedule a change request to roll out the new HSIJANLO started task to the production database host.
 - b. Copy the HSIJANLO started task from the JCLLIB data set to the system PROCLIB data sets.
 4. Configure the Analyzer utility to work with a secure socket layer (SSL) for HTTPS transport and to logon with a RACF user ID and password:
 - a. In the HSIANP2 member of the PARMLIB data set, change the security parameter to SECURITY=SYSTEM,
 - b. Review and edit the comments in the HSIANS1, HSIANS2, and HSIANS3 members of the JCLLIB data set to create a digital certificate that is required for SSL.
 - c. Configure the HTTPPORT parameter, if you require a value other than the default value.
 - d. Review the Analyzer reports to confirm that all expected products are identified.

Maintaining the production Repository database

You must perform regular maintenance tasks on the production Repository database to ensure that performance is optimal. The maintenance tasks cull obsolete and unwanted data and reorganize the database as necessary.

About this task

A database administrator or system programmer performs these maintenance tasks.

Procedure

1. Run the following jobs on a regular basis to delete old usage data, save space, and improve processing time:
 - a. Run the HSIUDEL job to delete usage data that are older than a specified period.
 - b. Run the HSIUSUM job to summarize usage data and compress records into monthly periods.
2. Run the HSI LDEL job to delete obsolete discovery and usage data for a specified system (LPAR).
3. Run the HSI STPRM job to reset the status flag back to normal for tables in the production Repository database, following a failure.

4. Run the following jobs on a regular basis to maintain the integrity and performance of data in the production Repository database:
 - a. Run the HSISUT01 job to backup the Repository database in DB2 or backup the zFS file system in SQLite.
 - b. Run the HSISUT02 job to restore the Repository database in DB2 or restore the zFS file system in SQLite.
 - c. Run the HSISUT03 job to reorganize the Repository database in DB2.
 - d. Run the HSISUT04 job to update Runstats statistics for the Repository database in DB2.

Chapter 3. Implementing deployment scenarios

Most implementations of Tivoli Asset Discovery for z/OS are based on one of the common deployment scenarios. An example is provided for implementing each of these common deployment scenarios with a DB2 Repository database. You can adapt an example for use with a SQLite Repository database.

Related concepts:

Chapter 1, “Planning for deployment,” on page 1

Before you deploy IBM Tivoli Asset Discovery for z/OS, consider which deployment option is best suited to your environment.

Scenario 1: Implementing a single Repository database with a single GKB database

The most common deployment scenario is an implementation with a single Repository database and a single global knowledge base (GKB) database.

About this task

The example deployment is for a DB2 database environment and includes the key parameters that influence this scenario.

Procedure

1. Customize an instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameters:
 - **DBTYPE**=DB2
 - **REPZSCHM**=TADZRE1
 - **GKBZSCHM**=TADZGK1
 - **DB**=TADZREP1
 - **DBGKB**=TADZGKB1
2. Submit the HSISCUST job.
3. Create the Repository and GKB databases and grant access to them:
 - a. Run the HSISDB01 job to create storage groups.
 - b. Run the HSISDB02 job to create the GKB database and database objects.
 - c. Run the HSISDB03 job to create the Repository database and database objects.
 - d. Run the HSISGKBL job to load GKB data.
 - e. Run the HSISGRNT job to grant DBADMIN access to Tivoli Asset Discovery for z/OS administrator.
 - f. Run the HSISGRTB job to grant SELECT access to database tables.
4. Collect Inquisitor and Usage Monitor data:
 - a. Run the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.
 - b. Run the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
 - c. Run the HSISUMON job on all z/OS LPARs to collect usage data.
5. Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).

6. Import Inquisitor and Usage Monitor data at the central site:
 - a. Run the HSISIQIM job to import Inquisitor data into the Repository database for each LPAR.
 - b. Run the HSISUIMP job to import Usage data into the Repository database for each LPAR.

Scenario 2: Implementing multiple Repositories with a shared GKB database

This deployment scenario implements two Repositories in a single DB2 subsystem that share a single global knowledge base (GKB) database. The advantage of sharing the same GKB is that you need only apply monthly updates to a single GKB database.

About this task

The example deployment is for two Repositories in the same DB2 subsystem to enable the Analyzer to browse both Repositories at the same time.

Procedure

1. Customize an instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameters:
 - **DBTYPE**=DB2
 - **REPZSCHM**=TADZRE1
 - **GKBZSCHM**=TADZGK1
 - **DB**=TADZREP1
 - **DBGKB**=TADZGKB1
2. Submit the HSISCUST job.
3. Create the Repository and GKB database and grant access to them:
 - a. Run the HSISDB01 job to create storage groups.
 - b. Run the HSISDB02 job to create the GKB database and database objects.
 - c. Run the HSISDB03 job to create the Repository database and database objects.
 - d. Run the HSISGKBL job to load GKB data.
 - e. Run the HSISGRNT job to grant DBADMIN access to Tivoli Asset Discovery for z/OS administrator.
 - f. Run the HSISGRTB job to grant SELECT access to database tables.
4. Collect Inquisitor and Usage Monitor data:
 - a. Run the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.
 - b. Run the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
 - c. Run the HSISUMON job on all z/OS LPARs to collect usage data.
5. Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).
6. Import Inquisitor and Usage Monitor data at the central site:
 - a. Run the HSISIQIM job to import Inquisitor data into the Repository database for each LPAR.
 - b. Run the HSISUIMP job to import Usage data into the Repository database for each LPAR.

7. Customize another instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameters:
 - **DBTYPE**=DB2
 - **REPZSCHM**=TADZRE2
 - **GKBZSCHM**=TADZGK1
 - **DB**=TADZREP2
 - **DBGKB**=TADZGKB1
8. Create the second Repository and grant access to it:
It is not necessary to run jobs to create and populate the GKB database in this step because the second Repository shares the GKB that you created in Step 2.
 - a. Run the HSISDB01 job to create storage groups.
 - b. Run the HSISDB03 job to create the Repository database and database objects.
 - c. Run the HSISGRNT job to grant DBADMIN access to Tivoli Asset Discovery for z/OS administrator.
 - d. Run the HSISGRTB job to grant SELECT access to database tables.
9. Collect Inquisitor and Usage Monitor data to add to the second Repository database:
 - a. Run the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.
 - b. Run the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
 - c. Run the HSISUMON job on all z/OS LPARs to collect usage data.
10. Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).
11. Import Inquisitor and Usage Monitor data at the central site:
 - a. Run the HSISIQIM job to import Inquisitor data into the second Repository database for each LPAR.
 - b. Run the HSISUIMP job to import Usage data into the second Repository database for each LPAR.

What to do next

Repeat steps 7 -11 for each additional Repository that you want to create, changing the values for the **REPZSCHM** and **DB** parameters for each new Repository.

Scenario 3: Implementing multiple Repositories with multiple GKB databases

This deployment scenario implements two Repositories in a single DB2 subsystem, each with its own global knowledge base (GKB) database. This deployment scenario is not common because you must apply monthly updates to each GKB database.

About this task

The example deployment is for two Repositories in the same DB2 subsystem to enable the Analyzer to browse both Repositories at the same time.

Procedure

1. Customize an instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameters:

- **DBTYPE=**DB2
 - **REPZSCHM=**TADZRE1
 - **GKBZSCHM=**TADZGK1
 - **DB=**TADZREP1
 - **DBGKB=**TADZGKB1
2. Submit the HSISCUST job.
 3. Create the first Repository and GKB database and grant access to them:
 - a. Run the HSISDB01 job to create storage groups.
 - b. Run the HSISDB02 job to create the GKB database and database objects.
 - c. Run the HSISDB03 job to create the Repository database and database objects.
 - d. Run the HSISGKBL job to load GKB data.
 - e. Run the HSISGRNT job to grant DBADMIN access to Tivoli Asset Discovery for z/OS administrator.
 - f. Run the HSISGRTB job to grant SELECT access to database tables.
 4. Collect Inquisitor and Usage Monitor data:
 - a. Run the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.
 - b. Run the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
 - c. Run the HSISUMON job on all z/OS LPARs to collect usage data.
 5. Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).
 6. Import Inquisitor and Usage Monitor data at the central site:
 - a. Run the HSISIQIM job to import Inquisitor data into the Repository database for each LPAR.
 - b. Run the HSISUIMP job to import Usage data into the Repository database for each LPAR.
 7. Customize another instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameters:
 - **DBTYPE=**DB2
 - **REPZSCHM=**TADZRE2
 - **GKBZSCHM=**TADZGK2
 - **DB=**TADZREP2
 - **DBGKB=**TADZGKB2
 8. Submit the HSISCUST job.
 9. Create the second Repository and second GKB database grant access to them:
 - a. Run the HSISDB01 job to create storage groups.
 - b. Run the HSISDB02 job to create the GKB database and database objects.
 - c. Run the HSISDB03 job to create the Repository database and database objects.
 - d. Run the HSISGKBL job to load GKB data.
 - e. Run the HSISGRNT job to grant DBADMIN access to Tivoli Asset Discovery for z/OS administrator.
 - f. Run the HSISGRTB job to grant SELECT access to database tables.
 10. Collect Inquisitor and Usage Monitor data for the second Repository database:
 - a. Run the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.

- b. Run the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
 - c. Run the HSISUMON job on all z/OS LPARs to collect usage data.
11. Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).
 12. Import Inquisitor and Usage Monitor data at the central site:
 - a. Run the HSISIQIM job to import Inquisitor data into the second Repository database for each LPAR.
 - b. Run the HSISUIMP job to import Usage data into the second Repository database for each LPAR.

What to do next

Repeat steps 7- 12 for each additional Repository and GKB database that you want to create, changing the values for **REPZSCHM**, **GKBZSCHM**, **DB**, and **DBGKB** parameters for each new Repository and GKB database.

Scenario 4: Collecting and transferring Inquisitor and usage data from remote sites

This scenario extends each of the deployment scenarios to collect data from remote sites and transfer the data back to the central site for processing.

Procedure

1. At the remote site, install the target libraries.
2. Customize an instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameter:
DBTYPE=REMOTE
3. Submit the HSISCUST job.
4. Collect Inquisitor and Usage Monitor data and transfer the files to the central site for processing:
 - a. Run the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.
 - b. Run the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
 - c. Run the HSISUMON job on all z/OS LPARs to collect usage data.
 - d. Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).

Scenario 5: Implementing in a sysplex environment

This deployment scenario is for a sysplex environment where the DASD is fully shared across all z/OS LPARs that belong to the sysplex. This special deployment is similar to the deployment scenarios 1, 2, or 3 but the implementation steps are slightly different. The reason for using this approach is to achieve operational efficiency by just processing a single z/OS LPAR within a sysplex.

About this task

The example deployment is for a DB2 database environment and includes the key parameters that influence this scenario. For this scenario, assume that the sysplex contains four z/OS LPARs: MVSA, MVSB, MVSC, and MVSD.

Procedure

1. Customize an instance of the HSISCUST member in the hsi.SHSISAMP data set with the following parameters:
 - a. DBTYPE=DB2
 - b. REPZSCHM=TADZRE1
 - c. GKBZSCHM=TADZGK1
 - d. DB=TADZREP1
 - e. DBGKB=TADZGKB1
2. Submit the HSISCUST job.
3. Create the Repository and GKB databases and grant access to them:
 - a. Run the HSISDB01 job to create storage groups.
 - b. Run the HSISDB02 job to create the GKB database and database objects.
 - c. Run the HSISDB03 job to create the Repository database and database objects.
 - d. Run the HSISGKBL job to load GKB data.
 - e. Run the HSISGRNT job to grant DBADMIN access to Tivoli Asset Discovery for z/OS administrator.
 - f. Run the HSISGRTB job to grant SELECT access to database tables.
4. Collect and import Inquisitor data for for all z/OS LPARs the first time:
 - a. Run the HSISINQZ job on all four z/OS LPARs to collect Inquisitor data: MVSA, MVSB, MVSC, and MVSD.
 - b. Transfer the collected Inquisitor data to the central site via file transfer protocol (FTP).
 - c. Run the HSISIQIM job to import Inquisitor data into the Repository database for each z/OS LPAR.
5. Collect and import Inquisitor data only for a single z/OS LPAR in subsequent scans:
 - a. Set PLX=Y in the Inquisitor HSISINQZ job.
 - b. Run the HSISINQZ job on the first z/OS LPAR, MVSA, to collect Inquisitor data.
 - c. Transfer the collected Inquisitor data to the central site via FTP.
 - d. Run the HSISIQIM job to import Inquisitor data into the Repository database for the z/OS LPAR MVSA only.
 - e. Repeat steps a - d for z/OS LPAR MVSA every time a new scan is required.
6. Collect and import Inquisitor data for UNIX for all z/OS LPARs:
 - a. Run the HSISINQU job on all four z/OS LPARs to collect Inquisitor data for UNIX.
 - b. Transfer the collected Inquisitor data for UNIX to the central site via FTP.
 - c. Run the HSISIQIM job to import Inquisitor data for UNIX into the Repository database for each z/OS LPAR.
 - d. Repeat steps a - c for each z/OS LPAR every time a new scan is required.
7. Collect and import Usage Monitor data:
 - a. Run the HSISUMON job on all z/OS LPARs to collect usage data.
 - b. Transfer the collected Usage Monitor data to the central site via FTP.
 - c. Run the HSISUIMP job to import Usage data into the Repository database for each LPAR.

Chapter 4. Migrating to IBM Tivoli Asset Discovery for z/OS, version 8.1

When you migrate to the latest version of Tivoli Asset Discovery for z/OS from an earlier version, you must convert existing data to be compatible with your new environment.

Migrating to Tivoli Asset Discovery for z/OS from an earlier version

You can upgrade to Tivoli Asset Discovery for z/OS, version 8.1 from version 7.5 or version 7.2. This release introduces support for SQLite database, and you can migrate to either a DB2 Repository database or a SQLite database.

Migrating from version 7.5 to Tivoli Asset Discovery for z/OS version 8.1 (DB2 database)

When you upgrade to Tivoli Asset Discovery for z/OS version 8.1 for DB2 database, you do not have to port any data from the Repository database. The migration tasks focus on defining new DB2 objects and dropping obsolete DB2 objects.

Before you begin

Make a backup of your Tivoli Asset Discovery for z/OS version 7.5 Repository database.

Make a backup or rename your JCLLIB and PARMLIB data sets.

About this task

Perform these migration tasks for every DB2 Repository in your Tivoli Asset Discovery for z/OS environment.

Procedure

1. In Tivoli Asset Discovery for z/OS version 8.1, make a copy of the HSISCUST member in the hsi.SHSISAMP data set and modify the following parameters:
 - a. Set the value of the new **DBTYPE** parameter to DB2.
 - b. Set the value of the new **SYS** parameter to the system where the Repository database is located.
 - c. Set the value of the **DB** parameter to the same value that is defined for your existing version 7.5 Repository database.
 - d. Set the value of the **DBGKB** parameter to the same value that is defined for your existing 7.5 Global Knowledge Base (GKB) database.
 - e. Set the value of the new **REPZSCHM** parameter to the same value that is defined for the **DB** parameter.
 - f. Set the value of the new **GKBZSCHM** parameter to the same value that is defined for the **DBGKB** parameter.
2. Submit the HSISCUST job.
3. Edit and update jobs in the JCLLIB library and parameters in the PARMLIB library if there are special site requirements.

4. Run the following migration jobs:
 - a. Submit the HSISMI76 job to add a new MODEL_CAPACITY column in the NODE_CAPACITY table. This job was introduced as PTF UA65570 in version 7.5. A condition code of 0 is expected. If the column has already been defined, expect a non-zero condition code.
 - b. Submit the HSISMI81 job to add new DB2 objects to the Repository database. A condition code of 0 is expected.
 - c. Submit the HSISMI82 job to populate records and also delete obsolete records in some Repository tables. A condition code of 0 is expected.
 - d. Submit the HSISMI83 job to drop obsolete DB2 objects from the Repository database. A condition code of 0 is expected.
 - e. Submit the HSISMI84 job to verify that the previous migration tasks have been successfully implemented. A condition code of 0 is expected.
5. Submit the HSISGKBL job to populate the GKB database.

What to do next

After migration, use the following approach to manage the implementation to the new version.:

- Continue to use existing version 7.5 Inquisitor fully-scanned files as inputs for the version 8.1 HSISIQIM Inquisitor Import job.
- Continue to use existing version 7.5 usage data files as inputs for the version 8.1 HSISUIMP Usage Import job.
- Configure APF authorization for the version 8.1 SHSIMOD1 load library
- When the version 8.1 Inquisitor scans and Usage Monitors are ready for use, you can run 8.1 operational jobs and you can discontinue version 7.5 tasks.

Migrating from version 7.5 to Tivoli Asset Discovery for z/OS version 8.1 (SQLite database)

Tivoli Asset Discovery for z/OS version 8.1 introduces support for SQLite database. When you migrate from version 7.5 you can port your data from the DB2 Repository to the SQLite database.

Before you begin

Make a backup or rename your JCLLIB and PARMLIB data sets. Before considering porting your data from the DB2 Repository to the SQLite database, refer to the Product Overview section on the limitations of using SQLite

Procedure

1. In Tivoli Asset Discovery for z/OS version 8.1, make a copy of the HSISCUST member in the hsi.SHSISAMP data set and modify the following parameters:
 - a. Set the value of the new **DBTYPE** parameter to **SQLITE**.
 - b. Set the value of the new **SYS** parameter to the system where the SQLite Repository database is located.
 - c. Set the value of the new **REPZSCHM** parameter to the name of the table owner for the SQLite Repository objects.
 - d. Set the value of the new **GKBZSCHM** parameter to the name of the table owner for the GKB objects.
 - e. Set the value of the new **SQLTZFS** parameter to the name of the zFS linear VSAM data set.

- f. Set the value of the new **SQLTPATH** parameter to the path of the USS directory.
2. Submit the HSISCUST job.
3. Edit and update jobs in the JCLLIB library and parameters in the PARMLIB library if there are special site requirements.
4. Submit the following jobs:
 - a. Submit the HSISDB01 job to define the zFS VSAM linear data set
 - b. Submit the HSISDB02 job to create the GKB database.
 - c. Submit the HSISDB03 job to create the Repository database.
 - d. Submit the HSISGKBL job to populate the GKB database.
5. Run the HSISUNLD job from the version 7.5 JCLLIB library to unload data from the version 7.5 DB2 Repository database.
6. In version 8.1, submit the HSISMI8Q job to load the unloaded data into the SQLite Repository.

What to do next

After migration, use the following approach to implement the new version:

- Continue to use the fully-scanned Inquisitor files from version 7.5 as inputs for the version 8.1 HSISIQIM Inquisitor Import job.
- Continue to use the usage data files from version 7.5 as inputs for the version 8.1 HSISUIMP Usage Import job.
- Configure APF authorization for the version 8.1 SHSIMOD1 load library.
- When the version 8.1 Inquisitor scans and Usage Monitor data files are ready for use, run the version 8.1 operational jobs and discontinue version 7.5 tasks.

Migrating from version 7.2 to Tivoli Asset Discovery for z/OS version 8.1 (DB2 database)

When you upgrade to Tivoli Asset Discovery for z/OS version 8.1 for DB2, you must define new Global Knowledge Base (GKB) and Repository databases. You can port version 7.2 usage data across to version 8.1 for DB2.

Procedure

1. In Tivoli Asset Discovery for z/OS version 8.1, make a copy of the HSISCUST member in the hsi.SHSISAMP data set and modify the following parameters:
 - a. Set the value of the new **DBTYPE** parameter to DB2.
 - b. Set the value of the new **SYS** parameter to the system where the Repository database is located.
 - c. Set the value of the **DB** parameter to the name of the Repository database.
 - d. Set the value of the **DBGKB** parameter to the name of the GKB database.
 - e. Set the value of the new **REPZSCHM** parameter to the name of the table owner of **DB** Repository tables.
 - f. Set the value of the new **GKBZSCHM** parameter to the name of the table owner for the **DBGKB** tables.
2. Submit the HSISCUST job.
3. Edit and update jobs in the JCLLIB library and parameters in the PARMLIB library if there are special site requirements.
4. Submit the following jobs to create the databases:
 - a. Submit the HSISDB01 job to define DB2 storage groups.

- b. Submit the HSISDB02 job to create the GKB database.
 - c. Submit the HSISDB03 job to create the Repository database
 - d. Submit the HSISGKBL job to populate the GKB database.
 - e. Submit the HSISGRNT job to grant DBADM access to databases.
5. Submit the HSISMI75 migration job to export usage data from the version 7.2 Repository database.
 6. Submit the HSISUIMP usage import job to import usage data from the previous step.

What to do next

After migration, use the following approach to manage the implementation to the new version:

- Configure APF authorization for version 8.1 of the SHSIMOD1 load library.
- Run version 8.1 of the HSISINQZ job on all z/OS LPARs to collect Inquisitor data.
- Run version 8.1 of the HSISINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
- Run version 8.1 of the HSISUMON job on all z/OS LPARs to collect usage data.
- Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).
- Run version 8.1 of the HSISIQIM job to import Inquisitor data into the Repository database for each LPAR.
- Run version 8.1 of the HSISUIMP job to import Usage data into the Repository database for each LPAR.

Data from version 7.2 of the Inquisitor and Usage files cannot be imported into a version 8.1 repository.

Migrating from version 7.2 to Tivoli Asset Discovery for z/OS version 8.1 (SQLite database)

Tivoli Asset Discovery for z/OS version 8.1 introduces support for SQLite database. When you migrate from version 7.2 you can port your data from the DB2 Repository to the SQLite database.

Procedure

1. In Tivoli Asset Discovery for z/OS version 8.1, make a copy of the HSISCUST member in the hsi.SHSISAMP data set and modify the following parameters:
 - a. Set the value of the new **DBTYPE** parameter to **SQLITE**.
 - b. Set the value of the new **SYS** parameter to the system where the SQLite Repository database is located.
 - c. Set the value of the new **REPZSCHM** parameter to the name of the table owner for the SQLite Repository objects.
 - d. Set the value of the new **GKBZSCHM** parameter to the name of the table owner for the GKB objects.
 - e. Set the value of the new **SQLTZFS** parameter to the name of the zFS linear VSAM data set.
 - f. Set the value of the new **SQLTPATH** parameter to the path of the USS directory.
2. Submit the HSISCUST job.

3. Edit and update jobs in the JCLLIB library and parameters in the PARMLIB library if there are special site requirements.
4. Submit the following jobs:
 - a. Submit the HSIADB01 job to define the zFS VSAM linear data set
 - b. Submit the HSIADB02 job to create the GKB database.
 - c. Submit the HSIADB03 job to create the Repository database.
 - d. Submit the HSIAGKBL job to populate the GKB database.
 - e. Submit the HSIAGRNT job to grant access to z/OS OMVS groups.
5. Submit the HSIAMI75 migration job to export usage data from the version 7.2 Repository database.
6. Submit the HSIUIMP usage import job to import usage data from the previous step.

What to do next

After migration, use the following approach to manage the implementation to the new version:

- Configure APF authorization for the version 8.1 SHSIMOD1 load library.
- Run version 8.1 of the HSIINQZ job on all z/OS LPARs to collect Inquisitor data.
- Run version 8.1 of the HSIINQU job on all z/OS LPARs to collect Inquisitor data for UNIX.
- Run version 8.1 of the HSIUMON job on all z/OS LPARs to collect usage data.
- Transfer the collected Inquisitor and Usage Monitor data to the central site via file transfer protocol (FTP).
- Run version 8.1 of the HSIQIM job to import Inquisitor data into the Repository database for each LPAR.
- Run version 8.1 of the HSIUIMP job to import Usage data into the Repository database for each LPAR.

Data from version 7.2 Inquisitor and Usage files cannot be imported into a version 8.1 repository.

Migrating from Tivoli License Compliance Manager for z/OS, version 4.2 to Tivoli Asset Discovery for z/OS, version 8.1

Tivoli License Compliance Manager for z/OS, version 4.2, customers must implement Tivoli Asset Discovery for z/OS, version 8.1, as a new install using either a DB2 database or SQLite database. Porting of Tivoli License Compliance Manager for z/OS version 4.2 Surveyor and Monitor data directly to Tivoli Asset Discovery for z/OS version 8.1 is not supported.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites 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 supply 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 Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Printed in USA