Tivoli Asset Discovery for z/OS
Version 7 Release 5

Administration Guide and Reference

SC27-2701-01

Tivoli Asset Discovery for z/OS
Version 7 Release 5

Administration Guide and Reference

SC27-2701-01

Note
Before using this information and the product it supports, read the information in "Notices" on page 261.

April 2011

This edition applies to Tivoli Asset Discovery for z/OS, Version 7, Release 5 (product number 5698-B39) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

About this document

The IBM® Tivoli® Asset Discovery for z/OS® Administration Guide and Reference explains how to install, configure, and use the product.

Who should read this document

This document is for system administrators who install and configure Tivoli Asset Discovery for z/OS, and for Software Asset Managers who perform daily system management tasks, such as asset discovery and reporting of software distribution.

What this document contains

The manual is divided into the following chapters:

At the back of the book the appendixes give information about the following topics:

What's new in Tivoli Asset Discovery for z/OS 7.5

Tivoli Asset Discovery for z/OS 7.5 contains the following updates to the product:

Usability enhancements:

- Simplified flows
- Simplified installation and configuration
- Simplified reporting

Performance improvements:

- Reduction in CPU usage and elapsed time.

Enhanced z/OS based reporting capability

- New z/OS based Analyzer package, enabling interactive web output by specifying the option to download to html, csv, txt or xls.

Due to the restructuring of this manual, individual changes have not been flagged with change bars.

How to read the syntax diagrams

The syntactical structure of commands described in this document is shown by means of syntax diagrams.

Figure 1 shows a sample syntax diagram that includes the various notations used to indicate such things as whether:

- An item is a keyword or a variable.
- An item is required or optional.
- A choice is available.
- A default applies if you do not specify a value.
- You can repeat an item.

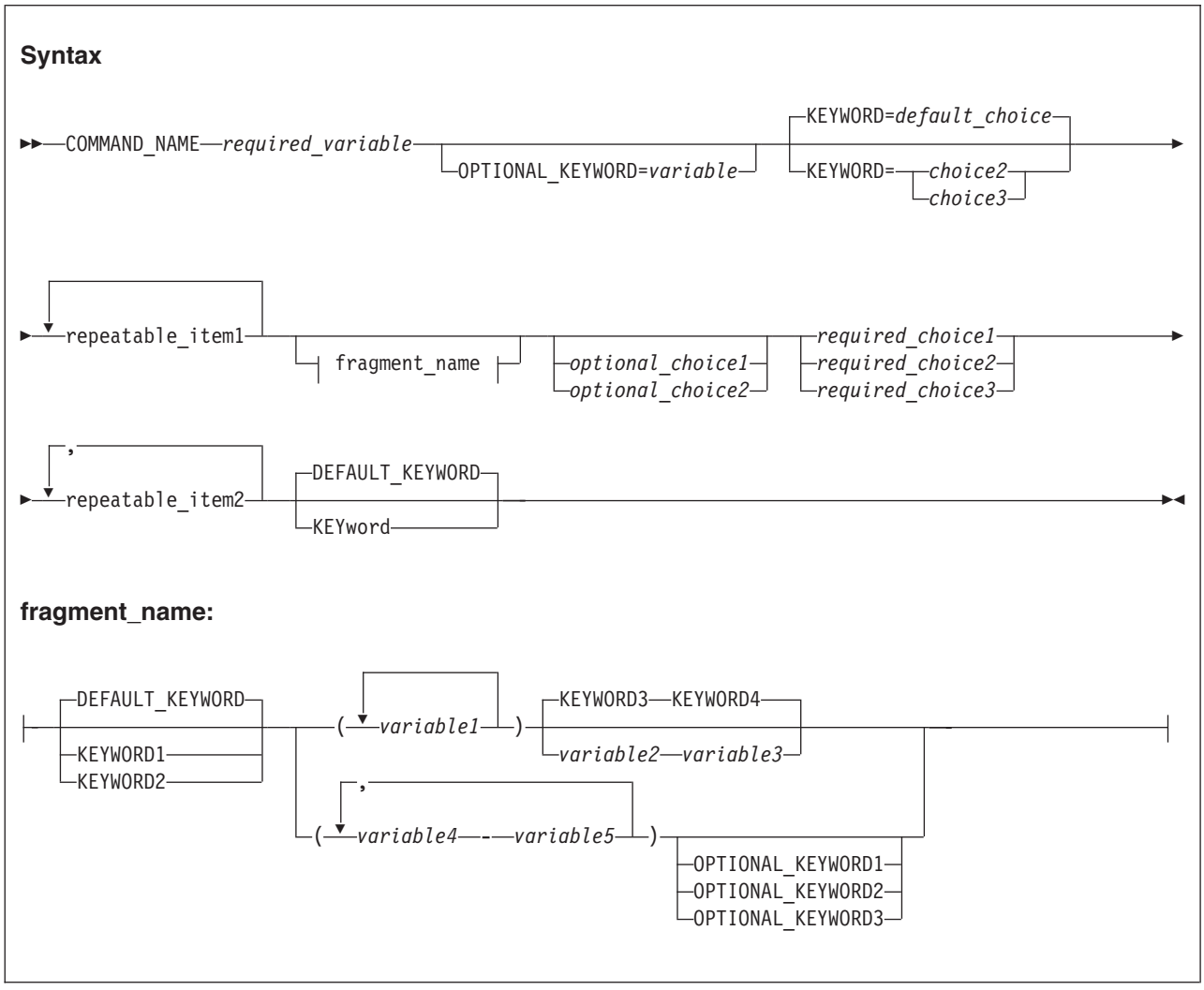


Figure 1. Sample syntax diagram

Here are some tips for reading and understanding syntax diagrams:

Order of reading

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

The ▶▶— symbol indicates the beginning of a statement.

The —▶ symbol indicates that a statement is continued on the next line.

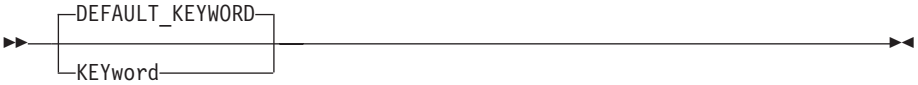
The ▶— symbol indicates that a statement is continued from the previous line.

The —▶◀ symbol indicates the end of a statement.

Keywords

Keywords appear in uppercase letters.

Sometimes you only need to type the first few letters of a keyword, The required part of the keyword appears in uppercase letters.



In this example, you could type "KEY", "KEYW", "KEYWO", "KEYWOR" or "KEYWORD".

The abbreviated or whole keyword you enter must be spelled exactly as shown.

Variables

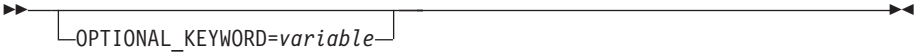
Variables appear in lowercase letters. They represent user-supplied names or values.

Required items

Required items appear on the horizontal line (the main path).

Optional items

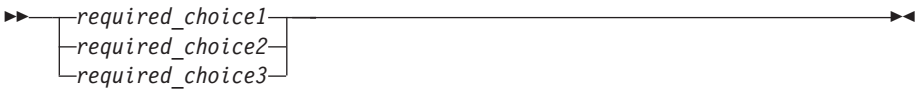
Optional items appear below the main path.



Choice of items

If you can choose from two or more items, they appear vertically, in a stack.

If you *must* choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.

If a default value applies when you do not choose any of the items, the default value appears above the main path.

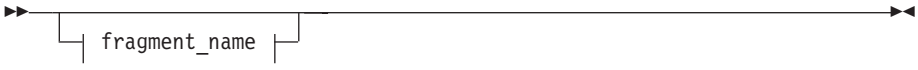
Repeatable items

An arrow returning to the left above the main line indicates an item that can be repeated.

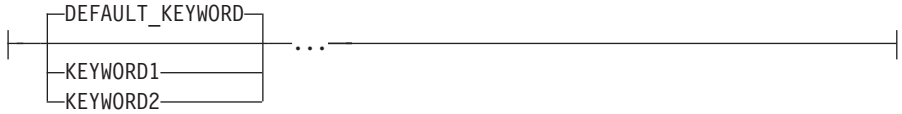
If you need to specify a separator character (such as a comma) between repeatable items, the line with the arrow returning to the left shows the separator character you must specify.

Fragments

Where it makes the syntax diagram easier to read, a section or *fragment* of the syntax is sometimes shown separately.



fragment_name:



Chapter 1. Overview of IBM Tivoli Asset Discovery for z/OS

Tivoli Asset Discovery for z/OS is built on the concept of remote and central mainframe components which work together to produce reports on z/OS mainframe products and their usage. This section provides you with a high-level overview of the Tivoli Asset Discovery for z/OS core architecture.

Tivoli Asset Discovery for z/OS runs on z/Architecture® mainframes that use the z/OS operating system. Its purpose is to:

- Discover and identify products for the z/OS platform.
- Monitor software usage and trends.
- Report on the MSU capacity of each LPAR under which the product runs.
- Provide reporting for assets and usage.

The benefits of using this software are:

- Used and unused software are identified.
- Users of software are identified.
- Obsolete versions of software are identified and the usage of these versions determined.
- Usage trends of software and libraries are identified.

In an IBM z/OS environment, software is contained in load libraries, or as z/OS UNIX files. The installed software products are determined by scanning the content of these libraries and files. The usage of the software products is determined by monitoring the loaded program files. The discovered load libraries and z/OS UNIX files are then checked against a global database of product information. This database is supplied with the product. Using this global database, the product infers which products are installed and used on a system. Usage events for the libraries and files that are part of a product are then accumulated as usage events for the product itself on the system upon which they occurred.

The workflow is illustrated in Figure 2 on page 2, followed by a brief description of the components.

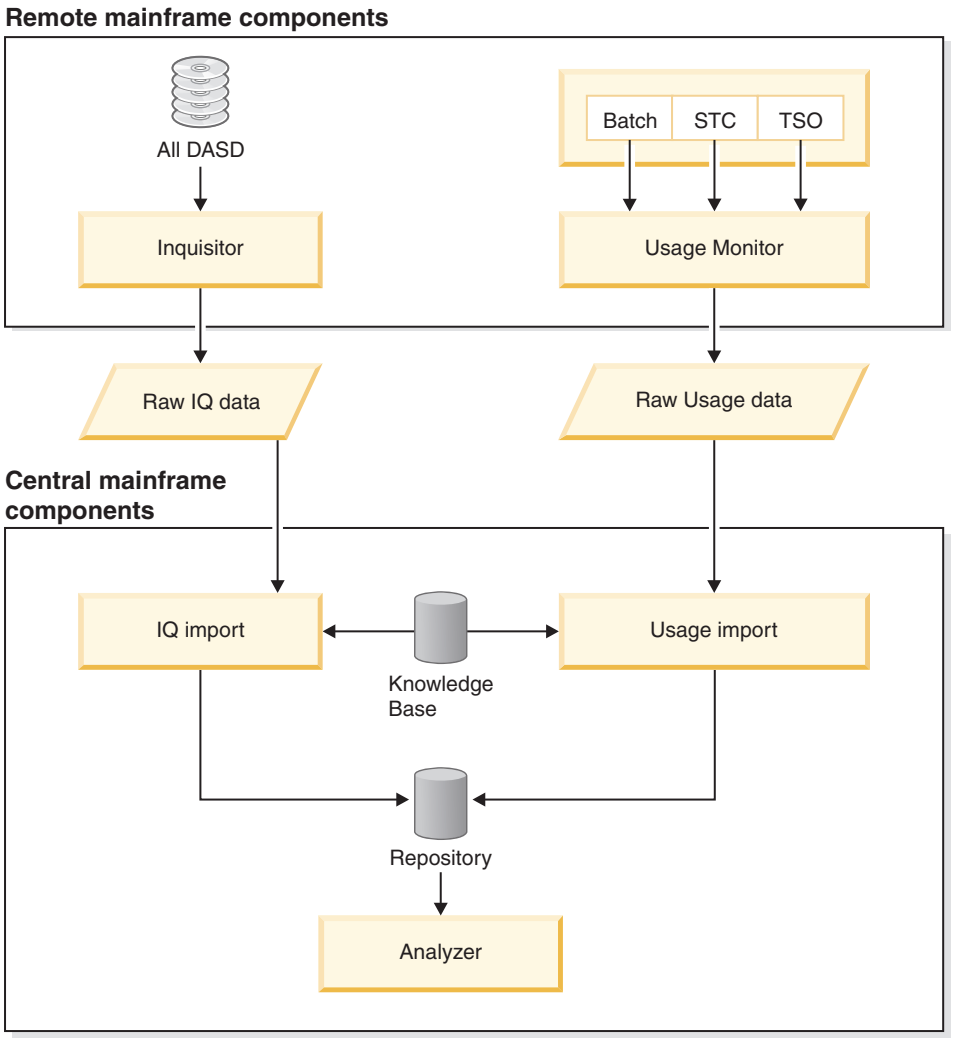


Figure 2. Product workflow

Inquisitor

A batch job which finds loadable programs in z/OS data sets and z/OS UNIX System Services file systems. One program locates load libraries on z/OS DASD devices, opens these load libraries, and captures information from the load modules in the libraries. This program is targeted to specific devices, libraries, or groups of libraries. It creates a compressed data set, which is then used as input to the Inquisitor Import process.

Another program locates and scans z/OS UNIX directories for program objects, and captures this information. It creates a compressed data set, which is then used as input to the Inquisitor Import for z/OS UNIX processing.

Usage Monitor

A started task which attaches to the z/OS program management functions by using standard IBM attachment protocols. It monitors and records the loaded modules of batch jobs, started tasks, and TSO users.

Usage Import

A batch job which copies the raw Usage Monitor data into the Repository

against matching load modules and aggregates the data against installed software products. After this process has been completed, you can view the usage data with the Analyzer reports.

Knowledge Base
The Global Knowledge Base, or GKB, is a database that is provided with Tivoli Asset Discovery for z/OS. It has a list of all z/OS globally identified products which are used by the product in the process of matching.

Inquisitor (IQ) Import
A batch job that loads raw Inquisitor data into IBM DB2® tables on z/OS, for z/OS load modules and z/OS UNIX program objects. The imported Inquisitor data is then matched against the Global Knowledge Base. The last step in the process is when the matched data is copied into the Repository.

Repository
A set of IBM DB2 for z/OS data tables that stores information about all of the software products discovered and their usage data.

Analyzer
The primary tool that is used to query the Tivoli Asset Discovery for z/OS database. It runs as a started task (STC), or batch job, on the same z/OS where the DB2 Subsystem runs. The output formats are HTML, Excel, Text, or Comma Separated Value (CSV). When configured in online mode, you can logon with your PC Browser; communicating directly with Analyzer on z/OS. The Analyzer can also be run in batch mode, where the results are saved to an output data set on z/OS.

(Optional) IBM Tivoli Common Reporting
A reporting tool, incorporating Cognos®, which can optionally be used to develop custom reports. This tool is useful if you want to combine data from Tivoli Asset Discovery for z/OS with other data sources. Logon with your PC browser, communicating with a distributed server where Tivoli Common Reporting is running, which then communicates with DB2 on z/OS.

Process flow

Data is collected on the target systems by the Inquisitor and the Usage Monitor, which are batch programs that do not require DB2 to be installed.

The collected data is then sent to the Repository database, where it is imported into the Repository tables. As the Repository is a set of DB2 objects, you must have DB2 installed where the Repository database is located.

In summary, the tasks as shown in Figure 2 on page 2 are:

1. Importing the data collected by the Inquisitor.
2. Matching the collected data with known product information from the Global and Local Knowledge Bases.
3. Loading the matched information into the Repository.
4. Importing the collected usage data into the Repository.
5. Running utilities to manage and maintain your data. This task is optional.
6. Reporting using the Analyzer, which consists of online and batch components.

   Optionally, you can use Tivoli Common Reporting Version 2 Release 1 to customize your reports.

Deployment

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

Inquisitor data
The Inquisitor scans DASD volumes for libraries containing load modules and HFS/zFS volumes for z/OS UNIX program objects, producing Inquisitor data. The Match Engine then associates these libraries and load modules with a Vendor, Product, Option, and Release. The Load to Repository then loads this information into the DB2 Repository. These processes are all performed by running the Inquisitor Import Job. See "Running the Inquisitor Import" on page 65.

Usage event
A usage event describes a unique load of a load module, or UNIX program object, for a particular Job, User, and, optionally, an Account Code. The Usage Monitor records these usage events as they occur on a particular operating system. When the usage data is imported into a Repository, the SMF ID of the system is used to associate the usage event with the load module discovered on that 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 DB2 tables that contains processed Inquisitor and Usage Monitor data. For all systems that are in a Repository, all DASD that is scanned must contain only unique volume serials; no duplicate SMF IDs. This means that any physically different volumes with identical volume names cannot be in the same Repository. Additionally, any physically different systems that are on a different OS/390® system, or LPAR, cannot coexist in the same Repository. Tivoli Asset Discovery for z/OS assumes that a library on a particular volume connected to one system is shared with other systems in the Repository that also contain that library and volume serial. The product also assumes that any SMF ID inside a Repository is unique, and any data that is imported from that SMF ID is to be merged with existing data associated with that SMF ID.

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:
There is a restriction that any DASD volumes with the same VOLSER are assumed to be the same volume and to be shared among all systems that have them mounted.

The only way to stop this sharing from happening is to place the systems involved into different Repositories. This means that you end up running one Repository for each Sysplex or stand-alone system. As a data center usually does not contain duplicate volume serials, or duplicate SMF IDs, it fulfills the criteria for a Tivoli Asset Discovery for z/OS Repository. It is common for IT service providers to define a separate Repository for each customer. It is also common practice for a specific client to have access to one or more systems. Defining a Repository around these separate systems also satisfies the need for separation of data and ease of

reporting. Additionally, when a Repository is scoped around a single client, it fulfills the criteria of not having duplicate volume names and unique SMF IDs within a Tivoli Asset Discovery for z/OS Repository.

It is recommended to have a central DB2 subsystem that contains all the Repositories in your entire enterprise. The usage and Inquisitor data that require processing should be transmitted to this centralized DB2 subsystem by using the Tivoli Asset Discovery for z/OS Automation Server or equivalent automation product. See "Automation Server for z/OS" on page 66.

Standard deployment

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 have scanned all available DASD on each system by running the Inquisitor, you can import all Inquisitor data from all systems into the relevant Repository by running the Inquisitor Import (see "Running the Inquisitor Import" on page 65). Tivoli Asset Discovery for z/OS analyzes this data and displays what products are installed on which systems. Usage data collected from every system by the Usage Monitor is imported and usage events are assigned to the discovered products so that you can analyze who has used a product on which system

The first step in deploying Tivoli Asset Discovery for z/OS is to create a deployment test repository. This 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 volumes and SMF IDs are unique.
- Determine database sizing.
- Analyze data integrity and correctness.

The next step is to run the SMP/E installation of the product, followed by the customization and creation of the required DB2 resources. 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 for a specific time period. A usual scenario is for this deployment test repository to be placed 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 data has been verified, it can be moved to a different subsystem with the appropriate backup strategies. However, in most cases this LPAR will not be a production LPAR.

Once you have moved your Repositories to their final location, you should consider setting up automation of the product. See "Automation Server for z/OS" on page 66.

Deployment for multiple Repositories

Some IT service providers will need to have multiple Repositories to cater for different sites. A site in this context means a logical split, for example, a data center, geographical region, or outsourcing customer. Having a Repository database for each site avoids the need to regularly transfer high volume usage data to a central site. It can also be easier to obtain DASD for several smaller databases than for one large database. The disadvantage of this set up is that reporting can only be performed on each specific Repository.

The amount of data a DB2 subsystem can accommodate is only limited by availability of disk space. For example, if your site has 50 or 100 z/OS systems, you can load all your data into a single Repository. However, for data management and performance reasons, it is recommended you consider grouping your data into logical units. You can define multiple Repositories under a DB2 subsystem, where each Repository might accommodate up to 30 systems. Each Repository must be defined with its own database. The advantage of this set up is that it makes it easier to generate reports across multiple Repositories.

You can implement multiple Repositories that share the same Global Knowledge Base. A shared Global Knowledge Base means that you only need to update a single copy of the Global Knowledge Base. Alternatively, each Repository can have its own Global Knowledge Base, but you will need to apply maintenance to each copy of the Global Knowledge Base.

Chapter 2. Installation

To set up Tivoli Asset Discovery for z/OS you need to perform a number of tasks. This section also details the prerequisites you need.

The tasks and prerequisites are described in the sections:

Quick set up

Setting up core components on a Test z/OS

Table 1. Setting up core components on a Test z/OS

Step	Description	Data sets and members
1	Install target libraries. A z/OS system programmer is normally required to complete this step. • Order or download the installation image for IBM Tivoli Asset Discovery for z/OS, Release 7.5 (PID 5698-B39), and all available maintenance. Use ShopzSeries or your usual IBM ordering process. • Using standard SMP/E for z/OS Receive and Apply, install as described in the *Program Directory, GI11-8965.* • Copy target libraries to the test z/OS LPAR where the DB2 for z/OS database is to be set up. • Dynamically define SHSIMOD1 to APF. For example, `SETPROG APF,ADD,DSN=hsi.SHSIMOD1,SMS` • Schedule a change request to roll out target libraries to all z/OS LPARs and include APF authorization for SHSIMOD1. For example, update appropriate PROG*xx* member.	hsi= TADz product prefix hsi.SHSIANL1 hsi.SHSIANL2 hsi.SHSIEXEC hsi.SHSIGKB1 hsi.SHSIMJPN hsi.SHSIMOD1 hsi.SHSIPARM hsi.SHSIPROC hsi.SHSISAMP hsi.SHSITCR1
2	Confirm DB2 prerequisites on the test z/OS LPAR. A DBA is normally required for this step. • DB2 for z/OS V8 (New Function Mode) or above. • DSNTIAD Dynamic SQL Program is enabled by using job SDSNSAMP(DSNTIJTM). • CLI/ODBC is enabled by using bind job SDSNSAMP(DSNTIJCL). • REXX is enabled by using bind job SDSNSAMP(DSNTIJRX), or SDSNSAMP(DSNTIJTM). • DB2 has access to a minimum of 4000 cylinders of 3390 DASD space.	DB2 SDSNSAMP data set members: • DSNTIJTM • DSNTIJCL • DSNTIJRX

Table 1. Setting up core components on a Test z/OS (continued)

Step	Description	Data sets and members
3	Define local environment settings. A DBA and a Tivoli Asset Discovery for z/OS Administrator to complete this step together. • Edit HSISCUST member in the SHSISAMP target library. Make changes within the SYSIN DD entry for local settings. Guidance is provided by documentation within the member. This job generates JCL jobs to be used in subsequent steps.	hsiinst=hlq for JCLLIB, and PARMLIB libraries hsi.SHSISAMP data set member: • HSISCUST generates hsiinst. &DB.JCLLIB hsiinst. &DB.PARMLIB
4	Create a Test system that contains new Storage Groups, GKB database, and the Repository database. • DBA submits HSISDB01 job. • DBA submits HSISDB02 job. • DBA submits HSISDB03 job. **Note:** A database on DB2 for z/OS is required on a central z/OS LPAR (NOT every LPAR).	JCLLIB data set member: • HSISDB01 • HSISDB02 • HSISDB03
5	Grant DB2 access. • DBA submits HSISGRNT job.	JCLLIB data set member: • HSISGRNT
6	Run Inquisitor DASD scan for z/OS product modules. A Tivoli Asset Discovery for z/OS Administrator to perform this step and all subsequent steps. • Generate Inquisitor data in data set pointed to by DD HSIPZIP. • For large sites, this job can take up to 1 hour to run. During this time you can continue with the subsequent tasks up to the Import Inquisitor data step. See "Running the Inquisitor" on page 31.	JCLLIB data set member: • HSISINQZ
7	Start the Usage Monitor • The Usage Monitor is commonly run as a started task, but as a quick test you can run it initially as a batch job. This job runs continually until it is manually stopped. Most of the time this job is idle. **Note:** Usage is not captured for address spaces that are already active before the Usage Monitor job has been started. See "Running the Usage Monitor" on page 49.	JCLLIB data set member: • HSISUMON
8	Load Global Knowledge Base (GKB) • Download latest GKB from IBM Support site, FixCentral, http://www.ibm.com/support/fixcentral. • Register for GKB update notification. • Load the GKB with HSISGKBL job. This process can take about 5 - 15 minutes, depending on your system capacity.	JCLLIB data set member: • HSISGKBL

Table 1. Setting up core components on a Test z/OS (continued)

Step	Description	Data sets and members
9	Import Inquisitor (IQ) data into the database. • Wait for Inquisitor job HSISINQZ to complete. • Submit HSISIQIM to import HSIPZIP data set created in Step 6 into the database. • For large sites, this job might take up to 2 hours to run the first time. Subsequent processing is about 90% quicker. See "Running the Inquisitor Import" on page 65.	JCLLIB data set member: • HSISIQIM
10	Import usage data. • This step is done by automation. However, as a quick test you can run it initially as a batch job. • Usage Monitor generates an output file `hsiinst.UM&SMFID.D*.T*` (which by default is automatically created at midnight), either by stopping the Usage Monitor, or issuing the `Switch z/OS modify` command. For example, `P HSISUMON or F HSISUMON,SW1` • Edit or submit HSISUIMP to import the Usage Monitor output data. See "Running Usage Import" on page 66.	JCLLIB data set member: • HSISUIMP
11	View the results with the Analyzer. • Review or change settings in PARMLIB(HSISANP1) • Submit JCLLIB(HSISANLO) on the same host as the database. This job runs continually until it is later manually stopped, for example `P HSISANLO` This job is idle for the remainder of the time. • Logon to the Analyzer with your PC browser. The default settings, which can be changed in PARMLIB(HSISANP1), are: – URL http://*hostname*:9000, where *hostname* is your z/OS IP host name or IP address. – User ID `tadzadm` – Password `tadz` • Examine the Analyzer reports and confirm that all expected products have been identified. If products are missing: – Check the GKB report to confirm the product is in the GKB. If it is not, open a PMR and IBM support will provide an updated GKB for the product. – If the GKB has the product, it might be missing because the product is not installed on the Test z/OS. In this case, you must run the Inquisitor on a system where the product is installed and import the Inquisitor data into the database. See "Running the Analyzer in online mode" on page 77.	PARMLIB data set member: • HSISANP1

Populating the Test database with discovery data

Table 2. Populating the Test database with discovery data

Step	Description	Data sets and members
12	Discover z/OS module-based products on all other z/OS LPARs. • Run HSISINQZ on other LPARs hsiinst.HSIPZIP.&SMFID. • Import the Inquisitor with member HSISIQIM. See "Running the Inquisitor Import" on page 65.	JCLLIB data set members: • HSISINQZ • HSISIQIM
13	Discover z/OS UNIX System Services based products on all other z/OS LPARs. • Run z/OS UNIX System Services Inquisitor. • Import hsiinst...HSIUZIP.&SMFID See "Running the Inquisitor for z/OS UNIX" on page 44.	JCLLIB data set members: • HSISINQU • HSISIQIM
14	Import Sub-Capacity Reporting Tool (SCRT) CSV data: • If your site has subcapacity licenses, the SCRT utility runs every month to scan SMF records. This utility generates a CSV file that is sent to IBM for billing purposes. The SCRT output CSV files can be imported into the Tivoli Asset Discovery for z/OS DB2 database to see trends, and combined with detailed data the product has captured. See "Running SCRT import utility" on page 98	JCLLIB data set members: • HSISSCRT
15	Convert Tivoli License Compliance Manager for z/OS Surveyor data into Inquisitor data. • Not applicable if you do not have Tivoli License Compliance Manager for z/OS.	JCLLIB data set members: • HSISS2D1 • HSISIQIM
16	Convert Tivoli License Compliance Manager for z/OS Monitor data into Usage Monitor data. • Not applicable if you do not have Tivoli License Compliance Manager for z/OS.	JCLLIB data set members: • HSISM2D1 • HSISUIMP
17	Optionally import historical usage data that has been collected by SMF • Run SMF Scanner utility to convert SMF type 30 records and SMF type 14 records into Usage Monitor data format. • Import the SMF Scanner output file. See "Running SMF scanner utility" on page 100.	JCLLIB data set members: • HSISSMF • HSISUIMP

Setting up automated data collection and import

Table 3. Setting up automated data collection and import

Step	Description	Data sets and members
18	Create a Production database cloned from the Test database. Most sites have a Test database (set up in previous steps) and a Production database (which runs on a Development LPAR, not a business workload LPAR). • The Production database must be set up to automatically import the captured usage data, either nightly or weekly (recommended for best performance). • Your database administrator should create the Production database by using the same process described in the steps that created the Test database. For example, `HSISCUST, HSISDB01, HSISGRNT` • Import the latest GKB with the HSISGKBL job. • Unload the Test database to a compressed file with the HSISUNLD job. • Initially load the Production database from the unloaded compressed file with HSISLOAD job. **Note:** The Production database has regular usage data imported. You can clone your Production database (with the usage data) and put it back into the Test database by performing an unload and load (in reverse). For example, HISISUNLD from Production, and HSISLOAD into Test.	JCLLIB data set members: • HSISCUST • HSISDB01 • HSISDB02 • HSISDB03 • HSISGRNT • HSISUNLD • HSISLOAD
19	Security. Using RACF® as an example: • Define a profile in the STARTED Class to associate a user ID to the started tasks HSIJMON, HSIJAUTO, and HSIJANLO • Ensure that user IDs have the following: – READ access to hsi** data sets – ALTER access to hsiinst.** data sets See “Security and authorization” on page 13.	
20	Usage Monitor started task. A z/OS system programmer is required to perform this step. • Review and Edit HSISMNPM settings. DSN(hsiinst.UM&SMFID) causes the output to generate hsiinst.UM&SMFID.D*.T* data sets. • Schedule change request to roll out new started task HSIJMON on all z/OS LPARs. • Copy HSISJMON to system PROCLIB data set. • Arrange for HSIJMON to be automatically started early in the IPL cycle to ensure that all activity is recorded. See “Setting up the Usage Monitor” on page 47.	PARMLIB data set member: • HSISMNPM JCLLIB data set member • HSIJMON

Table 3. Setting up automated data collection and import (continued)

Step	Description	Data sets and members
21	Automation Server started task. A z/OS systems programmer is required to perform this step. • Schedule change request to roll out new started task HSIJAUTO on all z/OS LPARs. • Define Automation control VSAM data set with HSIASALC job. • Customize HSIAPARM to run every weekend: – Remote hosts: Inquisitor scan, ZCAT to amalgamate, and FTP to transfer. – Database host: Inquisitor Import, Usage Monitor import, and aggregate. • Copy HSIJAUTO to system PROCLIB data set. • Arrange for HSIJAUTO to be automatically started. This task can be carried out any time in the IPL cycle. See "Running the Automation Server" on page 67.	JCLLIB data set member: • HSIASALC PARMLIB data set members: • HSIAPARM • HSAISCAN • HSIAZCAT • HSIAFTP • HSIAIQIM • HSIAUIMP • HSIAAGGR JCLLIB data set member • HSIJAUTO
22	Analyzer started task. A z/OS systems programmer is required to perform this step. • Schedule change request to roll out new started task HSIJANLO to the Production database host. • Copy HSIJANLO to system PROCLIB data sets. See "Getting started with the Analyzer" on page 75.	PARMLIB data set member: • HSISANP1 JCLLIB data set member: • HSIJANLO
23	Analyzer started task which uses Secure SSL (HTTPS) and logon with a RACF user ID and Password. A z/OS systems programmer and a RACF administrator are required to perform this step. In addition to the tasks described in Step 21, make the following changes: • Edit member HSISANP1 and change parameter to SECURITY=SYSTEM • Review and edit the comments in members HSISANS1, HSISANS2, and HSISANS3, to create a digital certificate required for Secure SSL. See "Analyzer JCLLIB and PARMLIB members" on page 76.	PARMLIB data set member: • HSISANP1 JCLLIB data set member • HSISANS1 • HSISANS2 • HSISANS3
24	Arrange for regular reorganization and backups of your database. A database administrator should carry out these tasks.	Only for Repository table spaces

Hardware requirements

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

- z/Architecture machine capable of running z/OS Version 1 Release 10 or later.
- (Optional) For Tivoli Common Reporting Version 2 Release 1, refer to Tivoli Common Reporting User Guide SC14-7613-00.

Software requirements

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

- z/OS Version 1 Release 10 or later.
- DB2 Version 8 Release 1 (new function mode) or later.

  Note: DB2 does not need to be installed on all your z/OS systems. However, the product needs DB2 to be installed on at least one z/OS system.
- Language Environment® for z/OS.
- Firefox Version 3.6.x, or Internet Explorer Version 6.
- (Optional) For Tivoli Common Reporting Version 2 Release 1, refer to Tivoli Common Reporting User Guide SC14-7613-00.

Security and authorization

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.

RACF customization

To run the Tivoli Asset Discovery for z/OS Started Tasks, Usage Monitor, Analyzer, and Automation Server, see the table shown here for the RACF authority required for each STC. Consult with your RACF Administrator to define the required RACF authority.

Table 4. RACF authority required for each started task

Started task name	SHSIMOD1	PARMLIB	SHSIANL1	SHSIANL2	ACDS	SDSNLOAD and SDSNEXIT	HLQIDS data set	Usage Monitor output database
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		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 *uuuuuuu* 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

For more information see “Security considerations when running the Inquisitor for z/OS UNIX” on page 46.

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.

DB2 authorization

You need DB2 privileges to perform the following tasks:

- DBADM authority to access the product database. You 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 and STATS privileges to run DB2 utilities LOAD and RUNSTATS.
- Access to work file database or TEMP database for Declared Global Temporary table.

Post-installation customization tasks for z/OS

This section describes the post-installation tasks you need to perform to customize the product in a z/OS environment.

A Tivoli Asset Discovery for z/OS administrator, and a DB2 for z/OS database administrator, should have the necessary authorities to implement these tasks.

z/OS customization

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

MAXCAD parameter

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, the Usage Monitor MAXCAD system parameter should be set to a value which is large enough to cater for the Usage Monitor data spaces. If specified, the MAXCAD parameter is located in the IEASYSxx member of the system PARMLIB library. For more information about the default and valid value range see the *MVS Initialization and Tuning Reference, SA22-7592* .

DB2 customization

A DB2 administrator should have the necessary authorities to perform the following tasks:

DB2 accesses

The product administrator needs to have DBADM access to the DB2 GKB and Repository databases.

For access to DB2 catalog tables for the Analyzer, see "Creating post-installation jobs" and read the definitions in HSISGRNT.

DSNTIAD Dynamic SQL Program

This product uses the DB2 supplied DSNTIAD program. To bind the plan and create the DSNTIAD program, use the job hlq.SDSNSAMP(DSNTIJTM).

Bind Call Level Interface (CLI) / ODBC support

The product uses the DB2 supplied DSNACLI plan. To bind the DSNACLI plan, customize the job hlq.SDSNSAMP(DSNTIJCL) and submit to run.

To avoid a common error associated with SQL error code -805 (DBRM OR PACKAGE NAME NOT FOUND IN PLAN), you must rebind this plan with the latest DB2 maintenance, and include the following package in the job:

```
BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIMS) -
 CURRENTDATA(YES) ENCODING(EBCDIC)
 SQLERROR(CONTINUE)
```

Bind REXX language support

This product uses the DB2 supplied DSNREXX plan. To bind the DSNREXX plan, use the job hlq.SDSNSAMP(DSNTIJRX) in DB2 version 8, and job hlq.SDSNSAMP(DSNTIJTM) in version 9, and later.

TEMP database

If you are using DB2 Version 8, at least one 8 K table space must be defined in a TEMP database to support Declared Global Temporary tables. This is not necessary for DB2 Version 9, or later, and can be done using the SQL:

```
CREATE DATABASE TEMPDB AS TEMP;
COMMIT;

CREATE TABLESPACE DSN8K01 IN TEMPDB
       USING STOGROUP SYSDEFLT
             PRIQTY 720
             SECQTY 144
             ERASE NO
       BUFFERPOOL BP8K0 SEGSIZE 4
       CLOSE NO   ;
COMMIT;
```

Creating post-installation jobs

To create a customized version of the post-installation JCLLIB and PARMLIB, take a copy of the job in member HSISCUST in the hsi.SHSISAMP data set and edit the job. This job contains parameters to define DB2 objects and jobs to run the product on the system. Contact your DB2 database administrator for assistance when you customize your DB2 parameters. For the SMS-related parameters, contact your system programmer.

Review the following table of parameters to customize the parameter values according to your requirements:

Parameter	Description
SET HSI	This JCL parameter must be set to the high-level qualifiers of the target libraries created by the SMP/E installation process. These data sets all have low-level qualifiers beginning with SHSI.

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. These libraries all have low-level qualifiers beginning with SISP.
DB	This parameter specifies the name of the DB2 Repository database that the product uses to store all the non-Global Knowledge Base information that it gathers. This parameter is used as a full qualifier for the tables and index definitions belonging to the Repository, and as a part qualifier for the tables and index definitions belonging to the Inquisitor, Local Knowledge Base, and Local Knowledge Base for z/OS UNIX. The DB name must be less than, or equal to, 8 characters in length.
HSIINST	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. The name specified for this parameter must be less than, or equal to, 19 characters in length.
CEERUN	This parameter specifies the fully qualified Language Environment data set.
CBCDLL	This parameter specifies the fully qualified Language Environment C++ runtime data set.
DB2LOAD	This parameter specifies the fully qualified SDSNLOAD data set name.
DB2EXIT	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.
DB2RUN	This parameter specifies the fully qualified RUNLIB data set name.
TIADPLAN	This parameter specifies the name of the DB2 plan used by the DSNTIAD utility. The DSNTIAD utility permits the use of dynamic SQL in a batch job, and is required by the post-installation jobs. The name of this DB2 plan can be obtained from your DB2 database administrator.
DBSSID	This parameter specifies the DB2 subsystem ID on the z/OS system.
LOC	This parameter specifies the ODBC(CLI) location for the DB2 subsystem ID on the z/OS system. You can use the DB2 DISPLAY DDF command to determine the location.

DBGKB	This parameter defines a single Knowledge Base database that is accessed by multiple Repositories under the same DB2 subsystem. It is used as part of the table qualifier and the index definitions qualifier belonging to the Global Knowledge Base, Global Knowledge Base for z/OS UNIX, and Inquisitor filters. 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.
DBADMIN	This parameter specifies the list of user IDs that are to be granted ADMIN access to the database and its contents. DBADMIN is optional. Specify an empty string if you do not want to grant ADMIN access to user IDs for the database specified in DB and DBGKB.
SETSQLID	This parameter specifies the DB2 Secondary Authorization ID, as used in SET CURRENT SQLID. This parameter is optional. Specify an empty string if you do not want to use DB2 Secondary Authorization IDs. The SETSQLID value must be less than, or equal to, 8 characters in length.
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.
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 239.
LOGGED	This parameter determines whether data changes to Inquisitor tables are recorded in the DB2 log. For DB2 Version 8, it must be set to `Y`. For DB2 Version 9 (new function mode) or later, you can improve performance by setting it to `N`. See Appendix D, "Performance and tuning," on page 239
SIZE	This parameter specifies the initial space allocations for DB2 table spaces of the 4 largest tables. The default value of SIZE is 1. See Appendix D, "Performance and tuning," on page 239.
CLASS	JES job class.
MSGCLASS	JES message class.
MSGLEVEL	JES message level.

Make the required changes to the copy of the customization job and submit it. The job creates or reuses two output PDSE libraries and one sequential data set. They are:

JCLLIB
The library contains JCL to implement and operate the product.

PARMLIB
The library is referenced by jobs in the JCLLIB, and contains pre-tailored parameters.

UM.HLQIDS
The sequential data set, referenced by the Usage Monitor when first created, contains a single record.

The JCLLIB members in the table are used to submit jobs to implement the product:

Table 5. Customized SET statements

HSISDB00	JCL SET statements to set symbols for data set high-level qualifiers. They are included by other jobs.

Table 6. Post-installation jobs

HSISDB01	Job to define DB2 storage groups.
HSISDB02	Job to define the Global Knowledge Base database.
HSISDB03	Job to define the Repository database.
HSISGKBL	Job to populate the Global Knowledge Base, Global Knowledge Base for z/OS UNIX, and Inquisitor filters.
HSISGRNT	Job to grant general access to DB2 catalog tables.

Table 7. Operations

HSISGKBL	Job to populate the Global Knowledge Base, Global Knowledge Base for z/OS UNIX, and Inquisitor filters. To be run when monthly updates are provided.
HSISINQZ	Job to run the Inquisitor.
HSISINQU	Job to run the Inquisitor for z/OS UNIX.
HSISUMON	Job to run the Usage Monitor.
HSISIQIM	Job to run the Inquisitor Import, Match Engine, Load to Repository for z/OS, and z/OS UNIX.
HSISUIMP	Job to run the Usage Import.
HSIASALC	Job to allocate the Automation Server control file.
HSIASSCT	Job to run the Automation Server Scout utility.

Table 8. Reporting

HSISANLO	Analyzer reporting in online mode.
HSISANLB	Analyzer reporting in batch mode.
HSISCOGA	Optional. Run this job only if you intend to use Tivoli Common Reporting V2.1 This job defines aliases for some Repository tables that are referenced by Tivoli Common Reporting V2.1
HSISCOGT	Optional. Run this job only if you intend to use Tivoli Common Reporting V2.1 This job defines DB2 catalog tables that are used for Tivoli Common Reporting V2.1. A REXX program is then run to populate the newly defined tables.

Table 9. Utilities

HSISZCAT	Job to concatenate and aggregate Usage Monitor data sets.
HSISPTAG	Job to run the Product Tagging utility.
HSISUSUM	Job to run the Usage Summary.
HSISUDEL	Job to run the Usage Deletion.
HSISLLST	Job to create an HLQ listing for the Usage Monitor.
HSISTPRM	Job to update the Repository TPARAM table.
HSISSCRT	Job to read SCRT CSV files and populate the Repository.
HSISKBT	Job to run the XML Export utility. **Note:** The XML output is for Tivoli Asset Management for IT.

Table 9. Utilities (continued)

HSISSMF	Job to get the historical usage information from existing SMF data.
HSISIBM	Job to filter out non-IBM programs from the Inquisitor and usage data.

Table 10. Jobs to unload and load data between Repositories

HSISUNLD	Job to unload the Repository tables.
HSISLOAD	Job to load the Repository tables.

Table 11. Globalization

HSISMCMP	Job to compile Japanese messages for MMS.

Running post-installation jobs

This section describes how to run your post-installation jobs once you have set them up.

If you are setting up a new system for a single Repository, submit all the post-installation jobs in the sequence listed in Table 12.

If you want to create multiple Repositories in the same DB2 subsystem, refer to the instructions described in steps 1 and 2 in "Migration tasks" on page 24.

Note: You need to APF authorize the SHSIMOD1 library for all systems that the product is to be run on.

If you have made changes to the JCLLIB and PARMLIB data sets, either manually or by rerunning the HSISCUST customization job, you need to redistribute the data sets to all the systems that run the product.

Table 12. Running post-installation jobs

Job name	Rerunnable	Acceptable Condition Codes (CC)	Comments
HSISDB01	YES	0 or 8	If storage groups have previously been created, then a CC of 8 is acceptable, otherwise investigate cause of failure.
HSISDB02	YES	0	If CC is not 0, then investigate cause of failure.
HSISDB03	YES	0	If CC is not 0, then investigate cause of failure.
HSISGKBL	YES	0 or 4	If CC is not 0 or 4, then investigate cause of failure.
HSISGRNT	YES	0 or 8	If grants for DB2 catalog tables have previously been run, then a CC of 8 is acceptable, otherwise investigate cause of failure.

Proceed to run the operation jobs, as described in Table 7 on page 19.

Verifying your installation

You can verify your installation by using the Analyzer to view the reports. If you have run the Inquisitor Import, the displayed output containing the Repository data tells you if you have been successful in your z/OS verification.

Chapter 3. Migration

This section describes how to migrate your data to the latest version of Tivoli Asset Discovery for z/OS if you are currently running one of the product versions listed here:

- IBM Tivoli Asset Discovery for z/OS V7.2.
- IBM Tivoli License Compliance Manager for z/OS V4.1, and V4.2

Migrating from V7.2 to V7.5

Before you migrate from V7.2 to V7.5, it is important to understand the improvements that have been made to the V7.5 database component.

1. There are no longer separate Inquisitor schema inventories in V7.5. The advantages of this change mean:
   - Less setup jobs.
   - Simpler structuring of Inquisitor schema inventories for shared and dedicated DASD pools.
   - Removes complexity of Usage Association.
   - Reduces database space.
   - Improves performance.

     Data is stored in the Repository for each unique module, library, and volume. In V7.2, this uniqueness was for each Inquisitor schema inventory, which could result in redundancy in the Repository. For example, one customer with 6 million modules had 300 million instances stored in their V7.2 database. In V7.5, they only had the 6 million instances.
2. Simplified processing jobs.
   - One job to import the Inquisitor data. This job includes steps to run the Match Engine and the Load to Repository.
3. Quicker processing.
   - Database component code is optimized to reduce processor consumption.
   - Inquisitor Import skips redundant processing.
     a. When processing freshly scanned Inquisitor data, the Repository is checked to see if there have been any changes to the library. By default, if the library content is the same, the Match Engine skips processing for that library.
     b. It is common for libraries to be cloned to different volumes and data set names. The Inquisitor Import detects cloned libraries and uses the information from the first time the library was matched in another library.

     This results in a significant reduction in the Inquisitor database processing time for repeat discoveries, making it possible to process fresh Inquisitor data weekly.
   - Usage Import REPMERGE step has been removed.

     The processing of REPMERGE is carried out internally in the raw Usage Import step, with optimized code for quicker processing.
   - The ZCAT utility optimizes weekly usage import processing.

The Usage Monitor collects usage statistics every day and, when these statistics are imported into the database, the information is stored in monthly granularity. The first and last day in the month the usage occurred is tracked.

ZCAT is enhanced to condense the daily usage to monthly granularity. This enhancement reduces the number of records the Usage Import must process. For example, if the same job runs every day of the week, 10 times a day, there would be one usage record in seven daily usage files, each with a usage count of 10. ZCAT condenses the seven records into one record, with a total usage count of 70. If the usage period spans two months, there is one record per month.

Benchmarks with large customer data have shown ZCAT condenses weekly data by an average of 75%. This means that the Usage Import only needs to process 25% of the number of records it did in V7.2.

Note: ZCAT condensing requires the Usage Monitor to sort the data. In V7.2, this sorting was controlled with the SRT setting and you had to set this to SRT=Y. In V7.5, the SRT setting is removed, and data is always sorted.

- The ZCAT utility and the Usage Monitor have optional settings to reduce the level of usage detail imported into the database. By default, usage is stored in the database for each unique combination of month, module, library, volume, job name, user ID, and job account code.
  - JNM=N causes usage not to be tracked per individual job name and instead is tracked by job type.
  - UID=N causes usage not to be tracked per individual user ID.
  - JAC=N causes usage not to be tracked per job account code.

  Benchmarks with large customer data show ZCAT condense weekly data with JNM=N by an average of 95%. The Usage Import needs only to process 5% of the number of records it did in V7.2.

  ZCAT has a ZCATDETL DD that can be used to archive the detailed information. For example, you can have Usage Monitor capture all the information, use ZCAT JNM=N to reduce that amount of data imported into the database, and ZCATDETL to archive the details. Using this process, the database is smaller. If you want to know the specific job names, you can review the ZCATDETL archive with the **Usage Monitor File Detail** report in the Analyzer.

Migration tasks

Due to the database changes that have been explained in detail, the steps to migrate from V7.2 to V7.5 are as follows:

1. Determine if you need multiple Repositories.

   Data is stored in the Repository for each unique module, library, and volume. Volume names are unique for all systems in a data center. If you are combining data from multiple data centers, you might have different physical volumes that have the same name and usage data. These volumes are merged if you import the data into the same Repository. For this reason it is preferred to have a Repository set up for each data center. The separate repositories can be in the same central DB2 Subsystem, or you can choose to have a DB2 Subsystem for each data center.

   IT Service Providers often have customer dedicated systems. Each customer environment is typically represented in its own Repository, so that reporting can be done for each customer environment without the risk of accidentally including data from other customer systems.

Note: The Analyzer supports multiple Repositories in the same DB2 Subsystem. You can select which Repository you want to report on. The **Product by Repository** report can also be used to compare Repositories.

Usage data can be imported into multiple Repositories concurrently.

2. Create a new V7.5 database for each Repository. This can be in the same DB2 Subsystem as the V7.2 database, or a different DB2 Subsystem.

   HSISCUST must be run for each Repository. In V7.5, the Repository database name is a middle qualifier for the JCLLIB and PARMLIB data sets, for example

   ```
   HSIINST.TADZDB1.JCLLIB
   ```

   Except for the Repository database DB setting, it is recommended you use the same HSISCUST settings for all Repositories in the same DB2 Subsystem.

   a. Set up the first Repository, for example TADZDB1
      - View HSISCUST in the SHSISAMP data set.

        Define the settings in HSISCUST, and submit to generate HSIINST.TADZDB1.JCLLIB and HSIINST.TADZDB1.PARMLIB data sets.
      - View HSIINST.TADZDB1.JCLLIB
        - Run HSISDB01 to create the Storage Group.
        - Run HSISDB02 to create the Global Knowledge Base.
        - Run HSISGKBL to load the Global Knowledge Base.
        - Run HSISDB03 to create the Repository TADZDB1.
        - Run HSISGRNT to grant access to the DB2 tables.

   b. Set up the second Repository, for example TADZDB2
      - View HSISCUST in HSIINST.TADZDB1.JCLLIB.
        - Change the DB setting for the second Repository (example DB = TADZDB2) and submit in order to generate HSIINST.TADZDB2.JCLLIB and HSIINST.TADZDB2.PARMLIB data sets.
      - View HSIINST.TADZDB2.JCLLIB
        - Run HSISDB03 to create the Repository TADZDB2.
        - Run HSISGRNT to grant access to the DB2 tables.

          Note: You do not run HSISDB01, HSISDB02 and HSISGKBL, as these jobs are for shared resources defined when the first Repository was set up.

   c. Repeat this process for each Repository.

3. Import Inquisitor data from every system into the appropriate Repository.

   The Inquisitor in V7.2 and V7.5 are similar, so it might be easier to capture the Inquisitor data with V7.2, and then import it into the V7.5 database. It is recommended that you initially scan every system. Systems that you think share all volumes might have some dedicated volumes. The only way to be sure is to scan each system. If the data scanned is shared, the V7.5 enhanced Inquisitor Import processing does not repeat the match processing for it, but tracks which systems have access to the library. After all the systems have been scanned and imported, you can review the Analyzer **Volume by System**, and **Library by System** reports to confirm which systems share all data. In cases where all data is shared, you can choose to scan just one of the systems by using the Inquisitor PLX=Y setting for future rediscovers. Alternatively, you can scan everything and the Inquisitor Import processing can determine what is shared.

4. Export usage data from your V7.2 Repository and import to V7.5.

- Run HSISMI75 job to export usage data from a V7.2 Repository. If you are splitting your V7.2 Repository into multiple V7.5 repositories, you must run multiple exports, specifying which systems to export by using the SIDLIST TPARAM setting. The exported data can then be imported into the V7.5 Repository with the standard V7.5 Usage Import job, HSISUIMP.

5. Import fresh usage data.
   - The Usage Monitor in V7.2 and V7.5 are similar, so you can import V7.2 captured usage data into the V7.5 database.

     Note: It is possible to use V7.5 ZCAT to condense data that has been concatenated with V7.2 ZCAT. The ZCAT72IN parameter causes ZCAT V7.5 to internally unconcatenate the input V7.2 data to work files, condense the data, and produce one output file.

IBM Tivoli License Compliance Manager for z/OS conversion

The conversion process allows you to manage when each system within the enterprise is to be converted from Tivoli License Compliance Manager for z/OS to Tivoli Asset Discovery for z/OS. It uses a staged approach which allows both products to run in parallel until you decide to cut over to Tivoli Asset Discovery for z/OS.

Tivoli Asset Discovery for z/OS and DB2 can be installed on the system of your choice, with Tivoli License Compliance Manager for z/OS continuing to run on the existing systems within the enterprise.

Once you have completed the conversion process, you can start using the Tivoli Asset Discovery for z/OS database and reporting capabilities against your entire enterprise, without having to install it on every system. It enables the data generated by your existing Tivoli License Compliance Manager for z/OS to be used by Tivoli Asset Discovery for z/OS.

How to run the conversion

In order to run the conversion, you need to follow these steps:

1. Ensure that you have already installed Tivoli Asset Discovery for z/OS. See Chapter 2, “Installation,” on page 7.
2. Customize Tivoli Asset Discovery for z/OS. See “Creating post-installation jobs” on page 15 and “Running post-installation jobs” on page 20.
3. Run the conversion jobs
   - HSISM2D - Monitor conversion of MONDETL data for Tivoli License Compliance Manager for z/OS V4.1 and V4.2.
   - HSISS2D1 -Surveyor conversion of SURVDATA data for Tivoli License Compliance Manager for z/OS V4.1 and V4.2.

Loading converted data into Tivoli Asset Discovery for z/OS

To load converted data into Tivoli Asset Discovery for z/OS, you need to run the following jobs:

1. HSISIQIM - Job to run the Inquisitor Import. Input data for this job is derived from output of HSISS2D1.
2. HSISUIMP - Job to run the Usage Import. Input data for this job is obtained from output of HSISM2D.

Your Monitor and Surveyor data are now successfully imported into Tivoli Asset Discovery for z/OS

For more information about the jobs that you have run, see "Creating post-installation jobs" on page 15.

Post-conversion tasks

- You can continue to run your existing Monitor and Surveyor as before. Tivoli Asset Discovery for z/OS, Usage Monitor, and the Inquisitor, can be run in parallel.
- When Tivoli Asset Discovery for z/OS has been fully implemented, and the new Usage Monitor and Inquisitor Scan are ready for use, the Tivoli License Compliance Manager for z/OS version of the Monitor and Surveyor can be discontinued, and no further conversion is necessary.

Tivoli License Compliance Manager for z/OS Exporter compatibility

To provide a set of functions like the ones used by the Exporter, four members are provided in the hsi.SHSIEXEC data set. The members query the information stored in DB2, and return output stored in the four Tivoli License Compliance Manager for z/OS Exporter DSECT formats.

Table 13. SHSIEXEC queries

REXX samples in SHSIEXEC	Tivoli License Compliance Manager for z/OS Exporter DSECT	Description
HSIIXPMO	XPMODS	Installed Load Modules
HSIIXPPR	XPPRODS	Installed Products
HSIIXPUM	XPUSAGEM	Load Module use
HSIIXPUP	XPUSAGEP	Product use

Some fields cannot be generated as they were in Tivoli License Compliance Manager for z/OS. These fields are marked with a question mark.

The description, function, and JCL required to run these samples is documented in each member.

The REXX samples TRNMODS, TRNUSGM, and TRNUSGP are provided with Tivoli License Compliance Manager for z/OS. They transform each record into transaction file format for import into a spreadsheet. If you run these records during migration, the date fields which are unavailable in Tivoli Asset Discovery for z/OS are displayed as '6F'; the hexadecimal EBCDIC representation of '?'

Before using these samples, ensure the DB2 REXX environment for the target DB2 system has been configured successfully.

DB2 resources affected by migration

To find out which DB2 resources are affected by application data migration, see Table 14. This table is provided as a reference, and definitions for these DB2 resources can be found in the PARMLIB members.

Table 14. DB2 resources affected by migration

Qualifier	Table Name	V7.2	V7.5
&DBGKB_GKB7	PRODUCT	WVDRGKB	WVDRGKB
&DBGKB_GKB7	TPARAM	WRULGKB	WRULGKB
&DBGKB_GKB7	TPRODLINK	WVDRGKB	WVDRGKB
&DBGKB_GKB7	TPRODUCT	WPDTGKB	WPDTGKB
&DBGKB_GKB7	TPRSMAP	WRULGKB	WRULGKB
&DBGKB_GKB7	TPTFFMID	WRULGKB	WRULGKB
&DBGKB_GKB7	TRULES	WRULGKB	WRULGKB
&DBGKB_GKB7	TSCORPAT	WSCPGKB	WSCPGKB
&DBGKB_GKB7	TVENDOR	WVDRGKB	WVDRGKB
&DBGKB_GKB7	TVERSION	WVERGKB	WVERGKB
&DBGKB_GKU7	TPARAM	WRULGKU	WRULGKU
&DBGKB_GKU7	TPRODUCT	WPDTGKU	WPDTGKU
&DBGKB_GKU7	TPTFFMID	WRULGKU	WRULGKU
&DBGKB_GKU7	TRULES	WRULGKU	WRULGKU
&DBGKB_GKU7	TSCORPAT	WSCPGKU	WSCPGKU
&DBGKB_GKU7	TVENDOR	WVDRGKU	WVDRGKU
&DBGKB_GKU7	TVERSION	WVERGKU	WVERGKU
&DBGKB_IQF7	TCOMPILERS	WIQFILTR	WIQFILTR
&DBGKB_IQF7	TIQFILTERS	WIQFILTR	WIQFILTR
&DBGKB_IQF7	TPARAM	WIQFILTR	WIQFILTR
&DBGKB_IQF7	TUXFILTERS	WIQFILTR	WIQFILTR
&DBGKB_IQF7	TXPCMODULES	WIQFILTR	WIQFILTR
&DBGKB_IQF7	TXPCSPEC	WIQFILTR	WIQFILTR
&DBGKB_IQF7	TXVENDORS	WIQFILTR	WIQFILTR
&DB_LKB7	TPARAM	WLOCLKB	WLOCLKB
&DB_LKB7	TPRODUCT	WLOCLKB	WLOCLKB
&DB_LKB7	TPTFFMID	WLOCLKB	WLOCLKB
&DB_LKB7	TRULES	WLOCLKB	WLOCLKB
&DB_LKB7	TSCORPAT	WLOCLKB	WLOCLKB
&DB_LKB7	TVENDOR	WLOCLKB	WLOCLKB
&DB_LKB7	TVERSION	WLOCLKB	WLOCLKB

Table 14. DB2 resources affected by migration (continued)

Qualifier	Table Name	V7.2	V7.5
&DB_LKU7	TPARAM	WLOCLKU	WLOCLKU
&DB_LKU7	TPRODUCT	WLOCLKU	WLOCLKU
&DB_LKU7	TPTFFMID	WLOCLKU	WLOCLKU
&DB_LKU7	TRULES	WLOCLKU	WLOCLKU
&DB_LKU7	TSCORPAT	WLOCLKU	WLOCLKU
&DB_LKU7	TVENDOR	WLOCLKU	WLOCLKU
&DB_LKU7	TVERSION	WLOCLKU	WLOCLKU
&DB_ZIQ	TCSECT	WC*	WCZIQTS
&DB_ZIQ	TDECISION	WD*	WDZIQTS
&DB_ZIQ	TLIBRARY	WL*	WLZIQTS
&DB_ZIQ	TMIGREPORT	WR*	WRZIQTS
&DB_ZIQ	TMODULE	WM*	WMZIQTS
&DB_ZIQ	TPARAM	WS*	WSZIQTS
&DB_ZIQ	TSYSTEM	WS*	WSZIQTS
&DB_UIQ	TDECISION	WDU*	WDUIQTS
&DB_UIQ	TLIBRARY	WLU*	WLUIQTS
&DB_UIQ	TMIGREPORT	WRU*	WRUIQTS
&DB_UIQ	TMODULE	WMU*	WMUIQTS
&DB_UIQ	TPARAM	WSU*	WSUIQTS
&DB_UIQ	TSYSTEM	WSU*	WSUIQTS
&DB	NODE	VAGGR	VAGGR
&DB	NODE_CAPACITY	VAGGR	VAGGR
&DB	PRODUCT	VAGGR	VPRODUCT
&DB	PRODUCT_INSTALL	VAGGR	VPRODINS
&DB	PRODUCT_NODE_CAPACITY	VAGGR	VAGGR
&DB	PRODUCT_USE	VAGGR	VPRODUSE
&DB	PRODUCT_USE_DETAIL	VSHARE	VPRODDET
&DB	SYSTEM	VAGGR	VAGGR
&DB	SYSTEM_NODE	VAGGR	VAGGR
&DB	TACCOUNT	VSHARE	VSHARE
&DB	TINVCTL	VSHARE	VSHARE
&DB	TINVREG	VSHARE	VSHARE
&DB	TIQHISTORY	VSHARE	VSHARE
&DB	TJOBDATA	VJOBDATA	VJOBDATA
&DB	TLIBRARY	VSHARE	VLIBRARY
&DB	TLIBSYS	N/A	VTLIBSYS
&DB	TLPAR	VSHARE	VSHARE

Table 14. DB2 resources affected by migration (continued)

Qualifier	Table Name	V7.2	V7.5
&DB	TMODULE	VMODULE	VMODULE
&DB	TPARAM	VSHARE	VSHARE
&DB	TPERIODS	VSHARE	VSHARE
&DB	TPOVINV	VSHARE	VPOVINV
&DB	TPOVLIB	VSHARE	VPOVLIB
&DB	TPRODUCT	VSHARE	VSHARE
&DB	TREGCLASS	VSHARE	VSHARE
&DB	TREGION	VSHARE	VSHARE
&DB	TREGLEAF	VSHARE	VSHARE
&DB	TUIMPORTCTRL	VSHARE	VSHARE
&DB	TUSELIB	VSHARE	VUSELIB
&DB	TUSEMTD	VUSEMTD	VUSEMTD
&DB	TUSEPO	VSHARE	VUSEPO
&DB	TUSEPOV	VSHARE	VUSEPOV
&DB	TUSEPOVLIB	VSHARE	VUPOVLIB
&DB	TUSEPRS	VSHARE	VUSEPRS
&DB	TUSERDATA	VSHARE	VUSRDATA
&DB	TVENDOR	VSHARE	VSHARE
&DB	TVERSION	VSHARE	VSHARE

Chapter 4. Operation

Knowledge Base updates

Updates to the knowledge databases are provided monthly. The GKB updates are available from the Fix Central website http://www.ibm.com/support/fixcentral. To get the updates from Fix Central, you must have a valid IBM user ID and password.

The following files are downloaded:

GKBLELVEyymmdd.TXT
The first part of this file has instructions similar to those in "Running the Knowledge Base updates." However, the instructions might include revisions to the procedure in this manual, or instructions that are specific to the accompanying update.

The second part of this file consists of a list of products affected by the change.

TADZKB
This file contains the replacement GKB to be loaded to the DB2 tables.

Running the Knowledge Base updates

The name of the file that contains the updates is TADZKB.XMI.

1. From the FixCentral website, http://www.ibm.com/support/fixcentral, download TADZKB.XMI as a binary file.
2. Upload TADZKB.XMI to the mainframe into a pre-allocated file with the attributes FB 80.
3. Receive the file by issuing TSO command `RECEIVE INDATASET(TADZKB.XMI).`
4. When prompted for additional information, enter `DA (name of file)`.
5. After receiving the file, run the Knowledge Base load job HSISGKBL. Before submitting this job, update SET INDSN= with the name of the file you have received.

If you want to be notified about Knowledge base updates by e-mail, proceed to the product web site http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Asset_Discovery_for_z~OS and click **Subscribe to this product** in the Notifications panel. Fill in the **My notifications** form and click **Submit**.

Inquisitor for z/OS

The Inquisitor is a program which scans and collects information about PDS and PDSE program libraries. The data collected by the Inquisitor is used as input to the Inquisitor Import, and forms the basis of your software inventory.

Running the Inquisitor

To run the Inquisitor, use the job HSISINQZ, that is found in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

Run time for this job depends on the number of volumes and libraries to be scanned. It is a good idea to run this job during off-peak periods.

When the HSISINQZ JCL is created, it has the ALLMSG, PLX=N, and LLQ=Z&SMF settings specified. If you want to alter these settings, use the parameters listed in the following table. The optional program parameter string lets you specify a report message level, a job run identifier, and an override to the system identifier. Use commas to separate the various settings specified within the program parameter string.

Table 15. Program parameter settings used for running the Inquisitor

Parameter	Description
DSNMSG	Requests that messages relating to processed data sets, which might otherwise be suppressed, are to be logged in the SYSPRINT report.
PGMMSG	Requests that messages relating to processed programs, which might otherwise be suppressed, are to be logged in the SYSPRINT report.
ALLMSG	Requests both DSNMSG and PGMMSG message logging.
NOAPF	Specifies that the Inquisitor is to run in an environment which is not APF authorized.
SID=	The value is up to 4 characters long, and specifies the system identifier to be contained in the data output from the Inquisitor. If the SID identifier override is omitted, the system SMF identifier is used. The SID parameter setting is used when the SMF system identifier of a system is not unique. For example: `SID=SYS2`
PLX=	The parameter is used to identify if the Inquisitor data being collected is part of a SYSPLEX. The value is either Y or N. If the PLX parameter is not used, the default value of N is created in the Inquisitor header record.
LLQ=	This parameter is used to specify a suffix string made up of one or more data set name qualifiers to be appended to the data set name of the HSIPZIP and HSIPOUT data set. Its maximum length is 44 characters. It may contain both static and dynamic system symbols, and the user symbols &SMF. (SMF system identifier) and &SYSLPAR. (LPAR name) supplied by the Inquisitor. Use the LLQ setting when you need to create uniquely named data sets without changing the JCL.

Table 16. Files used by the Inquisitor

File name	Description
SYSPRINT	A mandatory report file.
TAGREP	An optional report file that summarizes tag data collected by the Inquisitor.
SYSIN	A mandatory request input file. It processes fixed length, variable length, and undefined record formats. Records shorter than 72 bytes will be logically extended by the Inquisitor with blanks.
HSIPZIP	An optional output file that contains compressed Inquisitor data. It is written using a variable length record format. You must provide DCB information to ensure optimal use of DASD space.

Table 16. Files used by the Inquisitor (continued)

File name	Description
HSIPOUT	An optional output file that contains uncompressed Inquisitor data. It is not specified in the packaged sample, as the use of HSIPZIP is preferred, due to its reduced space requirements. HSIPOUT also contains variable length records. The program supplies the appropriate LRECL. By default, system determined block size is used. If you want to the direct the Inquisitor output to a compressible extended-format data set, then you should use the HSIPOUT file. The HSIPZIP file employs update-in-place processing, which prevents the use of DFSMS compression.
MCDS	An optional file that allocates the DFHSM MCDS data set, and is required if any requests contain the REMIGRATE or NOML2 operands. Further, if supplied for other requests, you can use it to avoid recalling data sets which are not load libraries. If the DFHSM MCDS is spread over more than one data set, use the DD names MCDS2, MCDS3, and MCDS4 consecutively. This allocates all the MCDS data sets in key range order.
ABRIN	An optional SYSIN file belonging to the FDRABRP utility program that is required if any requests contain the ABRMIG or ABRARC operands. It is primed by the Inquisitor during execution. For this reason, a single track VIO file is an ideal allocation.
ABRPRINT	An optional SYSPRINT file belonging to the FDRABRP utility program that is required if any requests contain the ABRMIG or ABRARC operands. It is an output-only file, and is not processed by the Inquisitor.

Table 17. SYSIN commands used by the Inquisitor

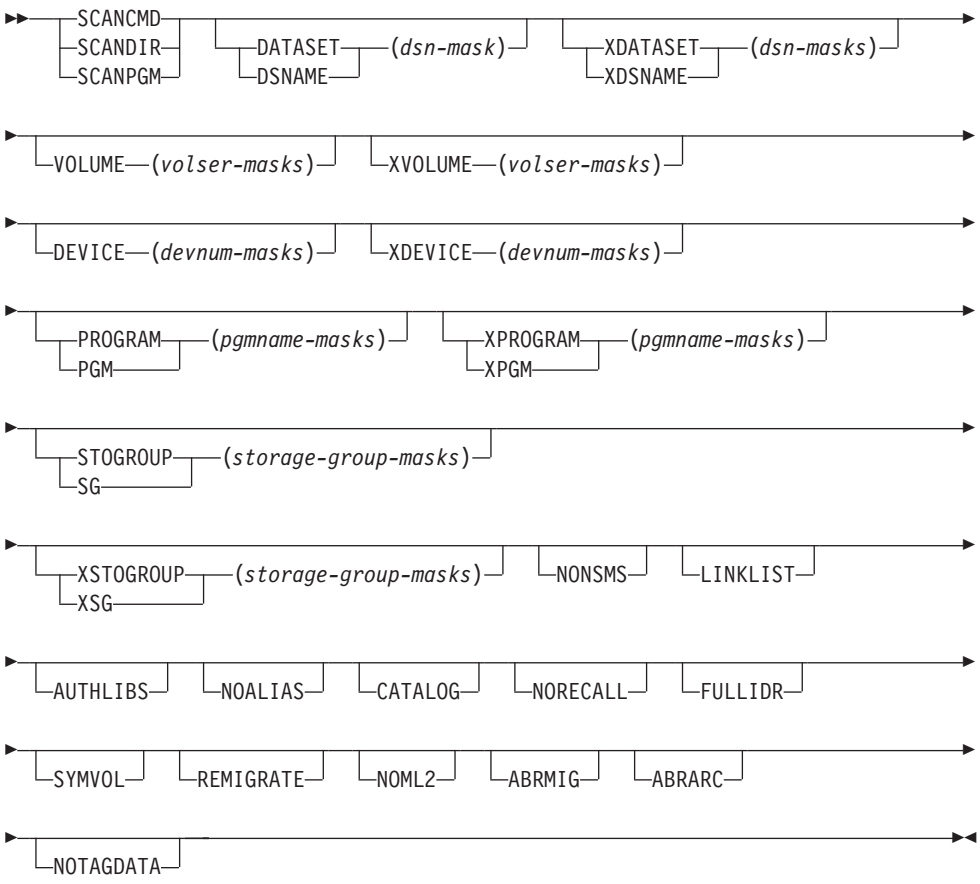
Parameter	Description
SCANCMD	Allows command syntax and operand consistency to be checked by the Inquisitor without initiating an actual scan for program libraries. It performs a parse only operation, although output files are opened. Error messages relating to syntax and operand errors are produced as usual. This verb is useful if you are formulating the best request combination when implementing on any given system.
SCANDIR	Collects data from program library directory entries. Contents of program members are not accessed. Compared to SCANPGM, its reduced data collection allows it to run faster. Although all syntactically correct operands are allowed, some operands relating to data from member contents are ignored during processing. SCANDIR collects all of the information needed for automated software identification, and is the command of choice for a production environment.
SCANPGM	Collects all data collected by SCANDIR, and information from member contents. Such information relates to program structure and history. Your IBM representative might request SCANPGM output data to assist with problem diagnosis and resolution.

Note:

The Inquisitor can process multiple requests in a single program run. The output of these requests is contained in the same file.

This syntax diagram shows the SYSIN commands and their operands.

Syntax diagram of Inquisitor commands



Operand defaults are:

```
DSNAME(*) VOLUME(*) DEVICE(*) PROGRAM(*)
```

All operands are optional. They are:

DATASET Alias: DSNAME

This operand specifies one or more 1 to 44 byte data set name masks. Only data sets with names matching any masks specified here are processed. Data sets with names not matching any masks specified here are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for selection matching. The precise treatment of asterisks in these masks is altered by the presence of the CATALOG keyword in the request. When CATALOG is specified, mask matching becomes qualifier aware and a single asterisk represents one, or part of, one qualifier only. When CATALOG is specified, use a double asterisk to specify any number of qualifiers. The data set name selection mask is the only mask affected by the CATALOG keyword. When the CATALOG keyword is present, exactly one DSNAME mask must be specified.

XDATASET Alias: XDSNAME

This operand specifies one or more 1 to 44 byte data set name masks. Data sets with names matching any mask specified here are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for exclusion matching. If this operand is used, each mask must specify a subset of a DATASET mask.

VOLUME

This operand specifies one or more 1 to 6 byte volume serial number masks. Only volumes with serial numbers matching any mask specified here are processed. Volumes with serial numbers not matching any mask specified here, are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for selection matching. A volume serial number mask of six asterisks specifies the current IPL volume, which is ascertained during execution.

XVOLUME

This operand specifies one or more 1 to 6 byte volume serial number masks. Volumes with serial numbers matching any mask specified here are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for exclusion matching. If this operand is used, each mask must specify a subset of a VOLUME mask. A volume serial number mask of six asterisks specifies the current IPL volume, which is ascertained during execution.

DEVICE

This operand specifies one or more 1 to 4 byte device number masks. Only volumes with device numbers matching any mask specified here are processed. Volumes with device numbers not matching any mask specified here, are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for selection matching. Standard character string mask matching is used. The use of characters which are not hexadecimal digits will not be detected by the program.

XDEVICE

This operand specifies one or more 1 to 4 byte device number masks. Volumes with device numbers matching any mask specified here are processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for exclusion matching. If this operand is used, each mask must specify a subset of a DEVICE mask. Standard character string mask matching is used. The use of characters which are not hexadecimal digits will not be detected by the program.

PROGRAM Alias: PGM

This operand specifies one or more 1 to 8 byte program name masks. Only programs with names matching any mask specified here are processed. Programs with names not matching any mask specified here, are not processed. Multiple masks must be separated by one or more delimiters.

This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for selection matching.

XPROGRAM Alias: XPGM
This operand specifies one or more 1 to 8 byte program name masks. Programs with names matching any mask specified here are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for exclusion matching. If this operand is used, each mask must specify a subset of a PROGRAM mask.

STOGROUP Alias: SG
This operand specifies one or more 1 to 8 byte storage group name masks. SMS-managed volumes in a storage group with a name matching any mask specified here are processed. SMS-managed volumes in a storage group with a name that does not match any mask specified here, are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for selection matching. Volumes which are not SMS-managed are not processed unless the NONSMS keyword operand is specified.

XSTOGROUP Alias: XSG
This operand specifies one or more 1 to 8 byte storage group name masks. SMS-managed volumes in a storage group with a name matching any mask specified here are not processed. Multiple masks must be separated by one or more delimiters. This operand can be specified more than once in a request, whereupon all masks specified in all occurrences of this operand are checked for exclusion matching. If both this mask and a STOGROUP mask are used, then each mask must specify a subset of a STOGROUP mask.

NONSMS
This keyword operand specifies that volumes which are not SMS-managed are eligible for processing. The presence of this operand means that SMS-managed volumes are not processed unless the STOGROUP operand was used to supply a storage group name mask.

LINKLIST
This keyword operand specifies that all link list data sets are to be unconditionally included for processing.

AUTHLIBS
This keyword operand specifies that all APF authorized data sets are to be unconditionally included for processing.

NOALIAS
This keyword operand specifies that any program member marked as an alias is to be excluded from processing.

CATALOG
This keyword operand specifies that data sets to be processed are located from a catalog search rather than VTOC searches. Data set alias names are not processed. The Inquisitor triggers and waits for a RECALL operation for each migrated data set which passes data set name mask processing, unless NORECALL is also specified.

NORECALL

This keyword specifies that migrated data sets are not to be recalled and are excluded from processing. This operand only has effect when the CATALOG operand is also specified. Data sets with a catalog entry indicating a volume serial number of MIGRAT, or ARCIVE, are deemed to be migrated.

FULLIDR

This keyword operand specifies that a full scan of CESD and IDR records is to be performed, even when a module would not have been selected for such processing. Depending upon the exact nature of the request being run, this operand can significantly elongate the elapsed time of Inquisitor runtime.

This operand is ignored for a SCANDIR request.

SYMVOL

This keyword operand specifies that when a load library resides on the IPL volume, or on a volume with a serial number which matches the value of a z/OS system symbol, then the output does not contain the actual volume serial number, but six asterisks for the IPL volume or the symbol name for other volumes. Only symbols with names which are six characters long, including the leading ampersand and excluding the trailing period, are considered for this processing.

If you use this keyword to collect your data, then you must also use the corresponding setting SYM(Y) in the Usage Monitor. You cannot use the SYMVOL parameter in the Inquisitor without using SYM(Y) in the Usage Monitor. By doing this, the Inquisitor data matches the usage data, otherwise the usage data is not imported to the correct products.

REMIGRATE

This keyword operand specifies that when a data set which had to be recalled has been processed, DFHSM is requested to migrate the data set again asynchronously. Migrated data sets can only be processed when the CATALOG operand is also specified. Only data sets with a catalog entry indicating a volume of MIGRAT are remigrated.

The presence of this operand requires that the MCDS file is allocated to the DFHSM MCDS. Access to the MCDS allows the Inquisitor to avoid recalls for data sets which are not partitioned, do not have an undefined record format, and do not have a block size of at least 1024.

NOML2

This keyword operand specifies that data sets migrated to level two are not to be recalled and are excluded from processing. Migrated data sets can only be processed when the CATALOG operand is also specified. Only data sets with a catalog entry indicating a volume of MIGRAT are checked for level two status.

The presence of this operand requires that the MCDS file is allocated to the DFHSM MCDS. Access to the MCDS allows the Inquisitor to avoid recalls for data sets which are not partitioned, do not have an undefined record format, and do not have a block size of at least 1024.

ABRMIG

This keyword operand indicates that when a catalog entry with a volume of MIGRAT is encountered, the FDRABR product is to be invoked to

determine whether a recallable archived copy of the data sets is available or not. If it is, then the data set is processed. If not, then the data set is not processed.

The NORECALL operand takes precedence over this operand.

The effect of ABRMIG is not affected by the ABRARC operand.

The presence of this operand requires that the ABRIN and ABRPRINT files are allocated.

ABRARC
This keyword indicates that, when a cataloged data set cannot be found on the volume, the FDRABR product is to be invoked in order to determine whether a recallable archived copy of the data set is available. If it is, then the data set is processed. If not, the data set is not processed.

The NORECALL operand takes precedence over this operand.

The effect of ABRARC is not affected by the ABRMIG operand.

The presence of this operand requires that the ABRIN and ABRPRINT files are allocated.

NOTAGDATA
This keyword indicates that data written to program libraries by the Product Tagger is not to be collected and written to the Inquisitor output file. Use this operand only when you do not want to update the Local Knowledge Base during the import process with the latest Tagger data that could be found by the Inquisitor.

SYSIN syntax rules for the Inquisitor

Syntax rules are as follows:

- Only the first 72 bytes of an input record are ever scanned.
- Short records are extended to 72 bytes with blanks.
- Blanks and commas are equivalent.
- Subparameters of value operands are specified in parentheses.
- A continuation to the next record is requested by a plus or a hyphen when it follows a delimiter, or is at the start of a record.
- A continuation cannot be requested in the middle of a word or value.
- The part of the record following a continuation character is ignored and can be used for comments.
- Records beginning with an asterisk are comment records.
- Records containing only blanks or commas are comment records.
- Comment records are ignored by syntax parsing logic, and do not alter continuation status.
- TSO conventions apply to abbreviations. That is, operands can be abbreviated to the minimum unambiguous length. Verbs cannot be abbreviated.
- If the input record contains an ampersand, the system symbol substitution routine ASASYMBM is called to perform symbol substitution processing.
- All input requests are parsed and stored before the first request is processed.
- If a syntax error is encountered, no requests are processed. This is to reduce the instance of incorrect or unproductive requests triggering lengthy DASD subsystem scans. The error is in the last record echoed in SYSPRINT.

- Value masks are character strings which are compared to data found at run time. Comparison is performed one byte at a time, from left to right. For a match, the characters must compare equal, unless a generic mask character is found.
- System static symbols, system dynamic symbols, and &SMF (SMF system identifier) and &SYSLPAR (LPAR name), can be used to construct value masks. &SYSLPAR may resolve to a null string if z/OS is running in a virtual machine.
- Valid generic mask characters are a percent (%), to flag a match for any single character, and an asterisk (*), to flag a match for any character string segment of zero or greater length.

Examples

These examples are provided to illustrate some possible scenarios where the scope and type of processing is customized.

Example 1: These three statements are equivalent, and request data collection for all programs on all online DASD volumes.

```
SCANDIR
SCANDIR DA(*) PGM(*)
SCANDIR VOL(*) DS(*)
```

Example 2:

To scan all SMS-managed volumes except volumes in storage group SGWORK use:

```
SCANDIR STOGROUP(*) XSTOGROUP(SGWORK)
```

Example 3:

To scan all volumes except volumes in storage groups with names beginning with SGW use:

```
SCANDIR XSTOGROUP(SGW*)
```

Example 4:

To scan all volumes with serial numbers beginning with TSO and WRK, these two requests are used in a single program run:

```
SCANDIR VOLUME(TSO*)
SCANDIR VOLUME(WRK*)
```

Example 5:

To scan all volumes except those with serial numbers beginning with TSO and WRK use:

```
SCANDIR XVOLUME(TSO* WRK*)
```

Example 6:

To scan all volumes with serial numbers beginning with USR which are also in SMS storage groups with names beginning with SG for programs with names beginning with UTIL, use: .

```
SCANDIR VOLUME(USR*) STOGROUP(SG*) PROGRAM(UTIL*)
```

Example 7:

To scan all data sets with high level qualifiers of SYS1, SYS2, SYS3, except z/OS distribution libraries, use:

```
SCANDIR DSNAME(SYS%.*) XDSNAME(SYS1.A*)
```

Example 8:

To restrict the data in the previous example to cataloged data sets, use:

```
SCANDIR DSNAME(SYS%.**) XDSNAME(SYS1.A*) CATALOG
```

Note: Note the extra asterisk in the data set name selection mask. Without this, only data set names with two qualifiers are selected. Data set name exclusion processing is not changed by the CATALOG operand.

Example 9:

To scan the current IPL volume, and any other link, list, and APF authorized libraries use:

```
SCANDIR VOLUME(******) LINKLIST AUTHLIBS
```

Example 10:

To scan the single cataloged data set SYS1.PPLIB without a lengthy DASD subsystem scan use:

```
SCANDIR DATASET(SYS1.PPLIB) CATALOG
```

Example 11:

To scan all cataloged SYS1 and SYS2 data sets use (a) two requests in a single program run, or (b) a single request. The two approaches exhibit similar resource consumption:

```
SCANDIR DA(SYS1.**) CAT
SCANDIR DA(SYS2.**) CAT

SCANDIR DS(SYS%.**) CAT XDSN(SYS3.*,SYS4.*,SYSA.*)
```

The XDSN values are coded as shown under the assumption that SYS1, SYS2, SYS3, SYS4 and SYSA are the only 4 character high-level qualifiers beginning with SYS on the system being scanned.

Note: SCANDIR DS(SYS1.**,SYS2.**) CAT is not allowed.

Example 12:

These examples are all equivalent. They scan the entire DASD subsystem for all data sets with a first qualifier of SYS1 or SYS2, excluding those with a second qualifier beginning with A.

(a)

```
SCANDIR DA(SYS1.*,SYS2.*) XDA(SYS1.A*,SYS2.A*)
```

(b)

```
SCANDIR DA(SYS1.*    +
SYS2.*)   +
XDA(SYS1.A*   +
SYS2.A*)
```

(c)

```
SCANDIR DA(SYS1.*)   +
DA(SYS2.*)   +
XDA(SYS1.A*)  +
XDA(SYS2.A*)
```

(d)

```
SCANDIR DA(SYS1.*)  XDA(SYS1.A*)  +
DA(SYS2.*)  XDA(SYS2.A*)
```

(e)

```
SCANDIR DA(SYS1.*)  XDSN(SYS1.A* SYS2.A*)  DS(SYS2.*)
```

Example 13:

Here, the entire DASD subsystem is processed, but the volume serial numbers are replaced in the output data by the name of a corresponding symbol, if one is defined, or by six asterisks for the IPL volume.

```
SCANDIR SYMVOL
```

The SYMVOL operand must be used in conjunction with the SYM command of the Usage Monitor otherwise the Usage data does not match the Inquisitor data.

Designing Inquisitor requests

When constructing statements for the Inquisitor SYSIN file, you need to ensure that a program library is never scanned more than once. A single Inquisitor request will not scan a VTOC or a library more than once. For this reason it is optimal to combine all selection and exclusion criteria to form a single SCANDIR request.

It can be difficult to formulate a system scan into a single CATALOG request, meaning that when the CATALOG operand is used, multiple requests are coded. Ensure that no data set will be scanned by more than one SCANDIR CATALOG request by excluding as many data set name patterns from each request as necessary. Data set name exclusions may not be necessary if all CATALOG search selection masks represent disjoint parts of the name space.

The example shown here uses the XDA operand to prevent SYS1.LINKLIB from being scanned more than once:

```
SCANDIR DA(SYS1.**) CATALOG
SCANDIR DA(SYS%.LINKLIB) XDA(SYS1.LINKLIB) CATALOG
```

As well as using the selection and exclusion facilities to ensure completeness, they can also be used to improve performance and efficiency by excluding DASD volumes which do not contain program libraries. Although a volume with no program libraries can be scanned quickly, processing duration might be reduced if such volumes can be excluded from an Inquisitor scan.

For example, volumes that only contain databases, or temporary data sets, do not have any files suitable for Inquisitor processing, but the VTOCs of those volumes are still read unless excluded by the appropriate selection criteria.

To illustrate this further, consider a system with these DASD subsystem usage elements:

System platform
: Non-SMS and storage group SYSTEM.

Work pool
Storage group TEMP containing temporary and short-lived (two days) permanent files.

TSO Storage groups TSOONE and TSOTWO.

Non-DB application
Non-SMS and storage groups BATCH1 and BATCH2.

Databases
Non-SMS volumes DBA001 to DBA099 and SMS storage groups DB01, DB02, and DB03.

The scanning of this configuration is to be carried out with the following assumptions:

- No need for data from libraries that do not exist for more than two days.
- No program libraries on database volumes.
- Applications combine their program libraries and non-database files.
- TSO users can have program libraries.
- Management requires information regarding all potentially permanent executable software.

To acquire Inquisitor data from all useful sources without processing volumes more than once, and without processing irrelevant volumes, you can specify multiple requests in a single Inquisitor run. For example:

```
SCANDIR SG(SYSTEM)
SCANDIR SG(TSO*)
SCANDIR SG(BATCH*)
SCANDIR NONSMS XVOL(DB*)
```

This can be consolidated into a single request giving the same result. For example:

```
SCANDIR SG(SYSTEM TSO* BATCH*) NONSMS XVOL (DB*)
```

Scanning migrated libraries

The Inquisitor locates load libraries by either scanning the VTOC of online volumes, or by searching the system catalog (CATALOG) for relevant data sets. When the keyword CATALOG is specified in a request statement, the Inquisitor passes the data set name selection mask to the Catalog Search Interface (CSI) to search for the catalog entries. It is possible that one or more of the catalog entries returned by the CSI are for a data set that has been migrated. In contrast, VTOC scans do not find migrated data sets.

Inquisitor processing of migrated data sets found by the CSI involves dynamic allocation which then triggers the recall of the data set. Recalls increase Inquisitor processing time. The processing leaves the data set in a recalled status.

The Inquisitor looks at the volume serial number in the catalog entry to determine if a data set is migrated or not. A data set is considered to have been migrated if its catalog entry indicates a volume serial number of either MIGRAT or ARCIVE.

To suppress the processing of all migrated data sets, specify the NORECALL keyword on each Inquisitor request.

Integration with DFHSM

If you are using the MCDS file allocation, and a data set cataloged on volume MIGRAT is encountered, the Inquisitor can read the data set record from the DFHSM Migration Control Data Set (MCDS) to verify that the data has the attributes of a program library. If the MCDS record is not found, the data set is ignored and processing is bypassed, avoiding a DFHSM error condition. If the data set does not have partitioned organization, an undefined record format, and a block size of at least 1024, the Inquisitor ignores the data set, avoiding the recall of many data sets which are not program libraries.

For systems with DFHSM space management functions, you can use the request keywords NOML2 and REMIG. The MCDS file allocation is a prerequisite for using the following keywords:

NOML2
Specifies that data sets migrated to level 2 are excluded from the scan.

REMIG
Specifies that after a recalled data set is processed by the Inquisitor, the Inquisitor requests DFHSM to remigrate the data set. The Inquisitor does not wait for the migration to complete, but begins to process the next data set immediately after making the request to DFHSM. Migration level 2 is never specified by the Inquisitor for the migration, even if the data set was recalled from ML2. (However, it might be selected by DFHSM as a result of SMS management class settings.)

Note:
Any combination of REMIGRATE, NOML2, and NORECALL is valid. Specifying NORECALL means NOML2 and REMIGRATE have no effect.

In the case where you want to scan all relevant migrated program libraries and do not want any such libraries explicitly remigrated afterward, you would not code any of the NORECALL, NOML2 and REMIGRATE keywords. In this instance, the MCDS file allocation, though optional, can still be used to great advantage.

Scanning generation data sets

Inquisitor CSI requests are limited to NONVSAM type A catalog entries. Generation data sets (which are members of a generation data group) are not scanned by Inquisitor CATALOG requests, though they can be processed by Inquisitor VTOC scans. Consider excluding generation data sets if you back up program libraries using generation data sets.

To exclude generation data sets from a VTOC scan request, specify a suitable data set exclusion mask, for example:

```
XDA(*.G%%%%V00)
```

Inquisitor for z/OS UNIX

The Inquisitor for z/OS UNIX collects information about executable software existing in HFS and zFS data sets currently mounted and accessible to z/OS UNIX. The data output by the Inquisitor is used as input to the Inquisitor Import, and forms the basis of your z/OS UNIX software inventory.

The Inquisitor for z/OS UNIX produces a set of record types which are different from those produced by the Inquisitor for z/OS. However, both programs collect the same types of information about installed software.

The Inquisitor for z/OS UNIX processes the HFS root directory, as well as all subdirectories. For this reason it needs to run with a UID which allows access to all directories and programs to be examined. If the Inquisitor for z/OS UNIX does not have permission to access a directory, then no information is collected from that directory, or any of its subdirectories.

The HSIXROOT file is used to nominate one or more directories to be considered root directories. When specified, only the nominated directories and their subdirectories are processed. This facility is useful when only a subset of the file hierarchy needs to be scanned.

The HSIXOMIT file is used to nominate one or more directories which are to be omitted or excluded from the scan, together with all of their subdirectories. This facility can be used to reduce resource consumption by preventing parts of the UNIX file hierarchy known not to have any executable software from being scanned.

Running the Inquisitor for z/OS UNIX

To run the Inquisitor for z/OS UNIX, use the job HSISINQU that is found in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

Run time for this job depends on the size and complexity of the UNIX directory structure to be scanned. It is a good idea to run this job during off-peak periods.

When HSISINQU JCL is created, it specifies the SYMLNK, PLX=N, and LLQ=U&SMF settings. If you want to alter this setting, use the parameters listed in the following table. The optional program parameter string of the Inquisitor enables you to specify a report message level, a job identifier, and override to the system identifier, and whether you want compressed or uncompressed output. If a program parameter is specified, it must start with a slash. Use commas to separate the various settings specified within the program parameter string.

Table 18. Program parameter settings used for running the Inquisitor for z/OS UNIX

Parameter	Description
PTHMSG	Requests that a message is written to HSIXMSG each time a directory is opened or closed.
PGMMSG	Requests that a message is written to HSIXMSG each time an executable file is processed.
ALLMSG	Requests both PTHMSG and PGMMSG message logging.
SYMLNK	Requests that symbolic links in the UNIX file system are to be processed in the scan. Use this parameter to improve the matching of UNIX program usage to identified software. Omit this to reduce UNIX file system scan time if you run the Usage Monitor with the USS(N) setting.
SID=	The value is up to 4 characters long and specifies the system identifier to be contained in the data output from the Inquisitor. If the SID identifier override is omitted, the system SMF identifier is used. The SID parameter setting is used when the SMF system identifier of a system is not unique. For example: `SID=SYS2`

Table 18. Program parameter settings used for running the Inquisitor for z/OS UNIX (continued)

Parameter	Description
PLX=	The setting is used to identify if the Inquisitor data being collected is a SYSPLEX. The value is either Y or N. If the PLX parameter is not used, the field is left blank in the Inquisitor header record.
OUT=	Specifies output file usage. The default value is Z. • A value of Z requests zipped output to HSIXZIP. • A value of T requests text output to HSIXOUT. • A value of B requests output to both HSIXZIP and HSIXOUT files.
LLQ=	This parameter is used to specify a suffix string made up of one or more data set name qualifiers to be appended to the data set name of the HSIXZIP and HSIXOUT data set. Its maximum length is 44 characters. It may contain both static and dynamic system symbols, and the user symbols &SMF. (SMF system identifier) and &SYSLPAR. (LPAR name) supplied by the Inquisitor. Use the LLQ setting when you need to create uniquely named data sets without changing the JCL.

Files used by the Inquisitor for z/OS UNIX

The files needed to run the Inquisitor for UNIX are listed in the following table:

Table 19. Files used by the Inquisitor for z/OS UNIX

File name	Description
HISIXMSG	Report file used by HSIXINQ.
SYSPRINT	Used by Language Environment (LE), which is required to be in the standard module search path, and by IDCAMS when LLQ= is specified.
SYSOUT	Used by Language Environment (LE), which is required to be in the standard module search path.
SYSIN	A mandatory request input file. It processes fixed length, variable length, and undefined record formats. Records shorter than 72 bytes will be logically extended by the Inquisitor with blanks.
HSIXZIP	An optional output file that contains compressed Inquisitor for z/OS data. It is written using a variable length record format. You need to provide DCB information to ensure optimal use of DASD space. The HSIXZIP file must never undergo any translation when being transferred, whatever the architecture of the target system. That is, only BINARY transfers are to be used to transport the file.
HSIXOUT	An optional output file that contains uncompressed Inquisitor for z/OS UNIX data. It is not specified in the packaged sample, as the use of HSIPZIP is preferred, due to its reduced space requirements. HSIXOUT also contains variable length records. The program supplies the appropriate LRECL. By default, system determined block size is used. If you want to the direct the Inquisitor for z/OS UNIX output to a compressible extended-format data set, then you should use the HSIPOUT file. The HSIPZIP file employs update-in-place processing, which prevents the use of DFSMS compression.

Table 19. Files used by the Inquisitor for z/OS UNIX (continued)

File name	Description
HSIXROOT	An optional file which can contain one or more records; each of which specifies a directory path to be considered as a root directory to be processed. If HSIXROOT is not allocated or empty, then a forward slash (/) is considered to be the only root directory to be processed.
HSIXOMIT	An optional file which can contain one or more records; each of which specifies a directory path which is to be omitted from the scan. Root directories cannot be omitted.

The HSIXROOT and HSIXOMIT files have the following characteristics and attributes in common:

- There is no requirement for the file to be allocated.
- The file might be empty or allocated to DUMMY.
- The file might contain fixed length or variable length records.
- Records must not contain more than 1024 bytes of data.
- Blank records are deemed to be comments and discarded.
- Leading and trailing blanks are discarded when the directory name is extracted.
- Records with an asterisk as the first nonblank are deemed to be comments and discarded.
- If the directory path does not end in a slash, then one is appended.

Security considerations when running the Inquisitor for z/OS UNIX

If you want to collect all relevant z/OS UNIX data, you must have access to all UNIX directories, including the root directory. This access ensures that all z/OS UNIX data is collected. To allow the Inquisitor unrestricted read access to all z/OS UNIX files, consider using the UNIXPRIV RACF Resource Class, which alleviates the need for UID(0).

The following sample definition can be used by your Security Administrator to define, permit, activate, and RACLIST the RACF UNIXPRIV Class:

```
RDEL UNIXPRIV SUPERUSER.FILESYS.**
RDEF UNIXPRIV SUPERUSER.FILESYS.** UACC(NONE) OWNER(IBMUSER)
PE  SUPERUSER.FILESYS.** CLASS(UNIXPRIV) RESET
PE  SUPERUSER.FILESYS.** CLASS(UNIXPRIV) ID(USERONE) ACCESS(READ)
SETR CLASSACT(UNIXPRIV)
SETR RACLIST(UNIXPRIV)
SETR RACLIST(UNIXPRIV) REFR
```

Usage Monitor

The Usage Monitor is a server address space which runs as a started task. Work is queued to it from all address spaces where programs are used. The Usage Monitor address space needs to be given a reasonably high dispatching priority.

Two different types of work can run in the server address space:

1. Moving of captured data into the data space repository. This is CPU bound, but of short duration.
2. Writing of the accumulated program usage data from the data space to a sequential file by the writer task. This is usually I/O bound.

The Usage Monitor runs APF authorized and is nonswappable.

Setting up the Usage Monitor

To configure the Usage Monitor you use the job that is found in the JCLLIB. This job is generated from the HSISCUST post-installation customization job. The name of this job is HSISUMON. This job will call the procedure HSIJMON from the JCLLIB.

Parameters for the Usage Monitor job are found in the PARMLIB member HSISMNPM.

Files used by the Usage Monitor

The Usage Monitor has three product-specific files. They are:

HSIZIN
A sequential file consisting of fixed length 80 byte records. It contains initial commands which are run before data collection becomes active. It must contain the data set prefix to be used for dynamically created output files. The prefix can be changed later by an operator MODIFY command.

HSIZIN is opened, read, and closed during initialization processing. Do not specify FREE=CLOSE in the JCL for HSIZIN, or refresh processing is not possible.

HSIZMSG
A log file which contains the initial commands issued, and which indicates their degree of success. It also contains regular status reports, refresh reports (when appropriate), and a termination report. It consists of fixed length 121 byte records.

SYSOUT
A report file used by the SORT program.

Output files containing program usage data are dynamically allocated by the Usage Monitor. The data set name prefix, the allocation unit, and the primary and secondary space allocation quantities (in tracks), need to be customized for the target system. This is done in the PARMLIB member HSISMNPM.

Using exclusion masks to reduce data

The data from a significant number of program usage events does not contribute meaningfully to the task of managing the software inventory. To reduce the processing of this unnecessary data, two mechanisms which allow some data to be excluded from collection have been provided. They are exclusion masking based on program name, and exclusion masking based on data set name.

Filtering program names:

A program name exclusion table exists which contains program name masks. When a program usage event is detected by the Usage Monitor, the program name is checked against entries in the program name exclusion table. When a match is found, the usage event data is discarded.

Each table entry contains a program name comparison string up to 8 bytes long. The string is either an 8 byte program name, or a shorter program name prefix. When entering these strings with the EXC command, a prefix is denoted by using an asterisk as the last character.

The program name exclusion table resides in key zero common storage, and its size is always a multiple of 4,096 bytes. The minimum table size can house up to 339 entries, and the table size increases dynamically, as required. The default program name exclusion table contains entries to exclude data pertaining to the usage of many programs which are part of the operating system.

In order to add, reset, remove, or display the entries to the table, use these commands:

EXC To add entries to the program name exclusion table, or to reset the table to its default contents.

DEL To remove some, or all, entries from the table.

D-X To display the current contents of the table.

Filtering a data set name:

Once the Usage Monitor has ascertained the name of the data set from which a used program was fetched, it is used to decide if the usage data is retained for collection or discarded. To perform this process, two lists of data set name masks are scanned; the first is the data set name inclusion mask list, and the second is the data set name exclusion mask list.

To avoid excessive storage and processor resource consumption, it is preferable to keep the number of elements in each list to a minimum. This is achieved by using generic masks to cover many data set names. The inclusion mask list is provided so that specific exceptions to broad exclusion rules can be specified. Unless data set name exclusion is to be used, the inclusion list cannot provide any useful function.

Elements of both lists reside in key zero common storage. Each element occupies 48 bytes, and contains a data set name mask up to 44 bytes in length. You can use the percent sign as a wildcard to match a single character. A trailing asterisk can be used to match the rest of the data set name.

In order to add, reset, remove, or display the entries to the tables, use these commands:

XDS To add a data set name mask to the exclusion list.

IDS To add a data set name mask to the inclusion list.

XDD To deactivate a data set name exclusion mask.

IDD To deactivate a data set name inclusion mask.

D-D To display the currently active masks in both lists.

Both lists have no elements until an XDS or IDS command is processed. Storage is dynamically acquired for each element as required. To ensure system integrity, XDD and IDD commands do not cause the storage of a deactivated element to be freed, but mark the element as inactive. When a deactivated mask is reactivated, the existing entry is marked as active without the further acquisition of storage.

When the Usage Monitor address space first initializes, all elements of both lists left in storage from a previous run are freed before the processing of initial commands and the commencement of data collection.

There is no requirement to use either data set name mask list at any stage.

Running the Usage Monitor

A Usage Monitor member named HSIJMON is provided in SHSIPROC. If you want to start HSIJMON as a started task, copy the customized member from the JCLLIB to an authorized PROCLIB.

To start the Usage Monitor in normal mode, issue:

```
S HSIJMON
```

Stopping the Usage Monitor

To stop the Usage Monitor, you can issue any of these commands:

```
P HSIJMON
F HSIJMON,STOP
F HSIJMON,END
```

These commands cause the Usage Monitor to stop data collection, attach a writer task to process the existing data in the data space, wait for the writer task to sort and output the data, and then terminate

QUICK Stop

For a quick stop, issue:

```
F HSIJMON,QUICK
```

This command causes the server address space to stop collecting data, attach a writer task to process the existing data in the data space, wait for the writer task to complete, and then terminate without sorting the data.

Immediate Termination

For an immediate termination, issue:

```
F HSIJMON,CAN
```

This command causes the server address space to stop data collection, detaches any running writer task which renders the output data set unusable, deletes the current data space without writing out its contents, and terminates.

Note: If you use the command `CANCEL` to stop the Usage Monitor, its data space is left in storage. To clear the data space from storage you must restart the Usage Monitor.

Refreshing Usage Monitor settings

The Usage Monitor has a range of commands to alter processing, any of which can be issued dynamically. However, these commands are only active for the duration of the current Usage Monitor session.

If you want to implement a change to both the running Usage Monitor, and to the initialization commands to be used by subsequent Usage Monitor sessions on startup, use the refresh facility.

Refresh processing involves the execution of the command stream placed in the HSIZIN file, without the requirement of stopping and restarting the Usage Monitor. As a result, refresh processing can verify the validity of the initialization

command stream so that changes are made and tested dynamically. This ensures that future Usage Monitor sessions do not encounter initialization command stream errors.

Some commands set a switch for logic control, or set a numeric value to be used during processing. These commands specify the values to be used in the future. Other commands pertaining to inclusion and exclusion masking add a mask to, or remove a mask from, the active mask list, so are part of an accumulation of commands which specify future processing.

Consider the example where several exclusion masks are active, and a change to deactivate one of the masks is required. A command to deactivate the mask might be issued dynamically, but if this change is to be made permanent, then the HSIZIN file needs to be updated. The alternative is to remove the command setting the exclusion from the HSIZIN file, and to then issue the Usage Monitor REF command to initiate a refresh.

Before the first HSIZIN command is run during refresh processing, the program mask exclusion list is set to the default list. Further, all data set name exclusion masks are deactivated, and all data set name inclusion masks are deactivated. This order of deactivation ensures that there is no loss of data that would otherwise be collected. However, there is the possibility that data which would have been excluded is collected during the short window between the reset of the mask lists and the processing of the HSIZIN commands.

The response to each command in the HSIZIN file is written to the HSIZMSG file. A summary WTO message, indicating whether any errors are found or not, is issued after refresh processing has finished.

Stopping the Usage Monitor and restarting it, produces the same active exclusion masks as a refresh. It also produces a data collection outage. For more information, see REF command in "Usage Monitor commands" for a list of the processes performed during a refresh operation.

Usage Monitor commands

The Usage Monitor commands are passed to the Usage Monitor from the HSIZIN input file, or by an operator MODIFY command.

The syntax rules are as follows:

- All commands are three characters long.
- Operands or subparameters are specified in parentheses.
- Multiple subparameters are separated by commas.
- The command must not contain any embedded blanks.
- Commands must start in column one.

To record the settings the Usage Monitor is using, place the display commands at the end of the HSIZIN file.

Details of each command follow.

CAP - Set hardware capacity collection status

CAP is used to specify if the Usage Monitor is to produce records

containing information about the hardware capacity of the system. Collecting this information is important when hardware capacity changes dynamically.

A change to this setting does not take effect until the next data space repository switch.

Y Specifies that hardware capacity data is collected and written out.

N Specifies that hardware capacity is not collected or written out.

If no CAP command is issued after IPL, the default is CAP(Y).

Example 1

Collect hardware capacity data.

```
F HSIJMON,CAP(Y)
```

Example 2

Do not collect hardware capacity data.

```
F HSIJMON,CAP(N)
```

CSA - Set the (E)CSA queuing storage limit

CSA is used to specify a limit to the quantity of (E)CSA storage used to queue work. If the Usage Monitor address space is not dispatched in a timely fashion, then many work elements can exist concurrently before being processed. Such work is queued in ECSA until it is transferred to the Usage Monitor repository.

If ECSA is exhausted, then CSA is used.

Data from program usage events occurring while this limit has been reached might not be collected.

An active CSA limit setting stays in force unless overridden, even if the Usage Monitor is stopped and restarted.

limit Specifies a number of kilobytes from 0 to 200,000.

If no CSA command is issued after IPL, the default is CSA(0). CSA(0) specifies that the Usage Monitor does not attempt to limit the (E)CSA storage used by work elements awaiting processing.

Example 1

Limit queuing in (E)CSA to 50,000 KB (almost 50 MB).

```
F HSIJMON,CSA(50000)
```

Example 2

Do not enable explicit (E)CSA limit for storing queued data.

```
F HSIJMON,CSA(0)
```

D-A - Display output allocation parameters

D-A is used to display dynamic allocation details to be used in the creation

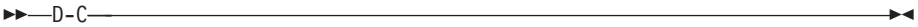
of output data files. The data set name, primary and secondary space quantities, and unit and optional volume serial number are shown.

Example
Display the current dynamic allocation values.

```
F HSIJMON,D-A
```

D-C - Display the counters and statistics

D-C is used to display the Usage Monitor activity and status indicators. The purpose of this command is to assist IBM technical support in problem diagnosis. The meaning of the output generated by this command is not published.



Example
Display the current value of internal Usage Monitor counters.

```
F HSIJMON,D-C
```

D-D - Display the data set name inclusion and exclusion lists

D-D is used to display the data set name masks in the inclusion list, followed by the data set name masks in the exclusion list.

The inclusion and exclusion lists do not need to be populated in order to collect data. The absence of any entries in the exclusion list means that data collection is not filtered by program library data set names.

Example
Display the current data set name inclusion and exclusion lists.

```
F HSIJMON,D-D
```

D-I - Display the system identifier

D-I is used to display the system identifier, which is written in the output header record. It can be altered by the **SID** command.

Example
Display the current system identifier used by the Usage Monitor.

```
F HSIJMON,D-I
```

D-S - Display the status settings

D-S is used to display several miscellaneous settings. Other commands are used to alter the individual settings, but this command provides a convenient way to list the current values.

Place at the end of the HSIZIN file to confirm monitoring settings.

Example

Display the current values of settings.

```
F HSIJMON,D-S
```

D-T - Display the automatic switch-and-write time setting

D-T is used to display the time-of-day specified for automatic data space switching and consequent writer task creation. When data from after this time-of-day is detected, data collection is automatically switched to a new data space, and write-out of data in the old data space is started.

The UTC or GMT switch time is calculated using local time current at data space creation time. The time when a data space is terminated is set when it is created. Changes to the system local time offset, such as those caused by a change to daylight saving time, do not alter the UTC or GMT that the current data space is closed. The time of the switch after the next switch is calculated using the new local time.

Example

Display the current automatic switch-and-write time setting.

```
F HSIJMON,D-T
```

D-X - Display the active exclude list

D-X is used to display the active program name mask exclude list. Data is not collected for programs with names that match the mask in any active entry in the exclude list.

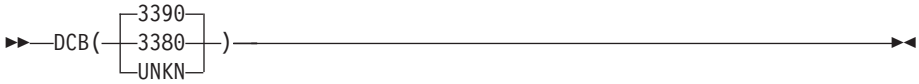
Example

Display the current exclude list entries.

```
F HSIJMON,D-X
```

DCB - Set output DCB attributes

DCB is used to set DCB attributes, which are optimal for a specific device type.



If no DCB command is issued, the default is DCB(3390).

DCB(3390)

Sets the output DCB to

```
RECFM=VB,LRECL=27994,BLKSIZE=27998
```

Use when the output device has 3390 compatible geometry.

DCB(3380)

Sets the output DCB to

```
RECFM=VB,LRECL=23472,BLKSIZE=23476
```

Use when the output device has 3380 compatible geometry.

DCB(UNKN)

Sets the output DCB to

```
RECFM=VBS,LRECL=32756,BLKSIZE=0
```

The system determines the optimal block size for the device used by dynamic allocation. Use when the output device type is not known until allocation time.

Some FTP products do not process a file with RECFM=VBS correctly, even when no records are actually spanned.

DEL - Deleting program mask entries

DEL is used to remove program name masks from filter tables. Both default and user-added entries can be removed. The required operand specifies one or more program name masks.

mask Specifies a 1 - 8 character program name mask. Any wildcard characters in the mask are treated as literals for the purposes of finding the mask to delete.

ALL Specifies every currently active mask. This mask cannot be specified with any other mask.

Except for short test periods, it is expected that default exclusion masks such as IGG* remain active.

Example 1

Remove all entries, so that all possible programs are monitored.

```
F HSIJMON,DEL(*ALL*)
```

Example 2

Remove exclusion masks to monitor LE and REXX modules.

```
F HSIJMON,DEL(CEE*,IRX*)
```

Example 3

Remove an exclusion mask to monitor the program called CEE.

```
F HSIJMON,DEL(CEE)
```

DSN - Setting the data set name prefix

DSN is used to specify the first part of the data set names used for the output files. The prefix is specified in the required operand.

Symbols are used in the construction of the data set name prefix. Available symbols include all z/OS static symbols, &SMF, the SMF identifier for the system, and &SYSLPAR, the logical partition name for the system.

dsnpref Specifies a 1 - 26 character data set name prefix. It can contain one or more data set qualifiers, and must not end in a period after any symbol substitution.

Note: Usage Monitor needs RACF ALTER access to the data sets to be able to create them.

Example

To get output files with names of the form

```
SYS3.HSI.HSIJMON.Dyyyyddd.Thhmmsst
```

use

```
F HSIJMON,DSN(SYS3.HSI.HSIJMON)
```

EXC - Adding program mask exclusion entries

EXC is used to add program name masks to the exclusion table. The required operand specifies one or more program name masks.

mask Specifies a 1 - 8 character program name mask. If the mask ends in an asterisk only, characters before the asterisk are compared. Otherwise, an exact program name is deemed to have been specified.

DFLT

Specifies every supplied default entry in the exclusion table is to be made active, and all user-added entries are to be removed from the exclusion table. This mask cannot be specified with any other mask.

Except for short test periods, it is expected that default exclusion masks such as IGG* would remain active.

Example 1

Reset the exclusion table to its default status.

```
F HSIJMON,EXC(*DFLT*)
```

Example 2

Exclude the collection of data for Language Environment modules and REXX modules.

```
F HSIJMON,EXC(CEE*,IRX*)
```

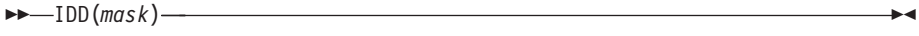
Example 3

Exclude the collection of data for the program CEE.

```
F HSIJMON,EXC(CEE)
```

IDD - Deleting data set name inclusion entries

IDD is used to remove data set name masks previously added by the **IDS** command.



mask Specifies a 1 - 44 character data set name mask. Any wildcard characters in the mask are treated as literals for the purposes of finding the mask to delete.

Example

Deactivate the SYS3.LINKLIB inclusion mask.

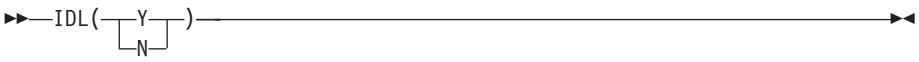
```
F HSIJMON,IDD(SYS3.LINKLIB)
```

IDL - Control idle work element usage

IDL is used to control whether the Usage Monitor uses idle work elements.

When the data in a work element has been processed, the element is normally freed in order to return the storage to the system. Enabling idle elements means that processed elements are retained on the idle chain. This chain is used before acquiring more storage when a new work element is needed.

Enabling idle elements reduces system storage management cost. The storage used by idle elements is included in the storage limit set by the **CSA** command.



Y Specifies the Usage Monitor retains processed elements for reuse, subject to the CSA limit setting.

N Specifies that all processed work elements are to be freed.

If no IDL command is issued after IPL, then the idle chain is used. IDL(Y) is the default setting.

IDS - Adding data set name inclusion entries

IDS is used to supply data set name masks, which specify data set names to be excluded from exclusion processing. Program usage data fetched from data sets with names matching inclusion masks, is collected without reference to the data set name mask exclusion list.

Inclusion masks are only useful if there are active exclusion masks. An inclusion mask is normally expected to match a subset of data set names, which would match an exclusion mask.

mask Specifies a 1 - 44 character data set name mask. If the mask ends in an asterisk only characters before the asterisk are compared. Percent signs in the mask indicate that any character in that location is considered a match.

Example If your intention is to not collect program usage data for data sets with a high-level qualifier of SYS3, except for SYS3.LINKLIB. SYS3.LINKLIB is the only data set with a high-level qualifier of SYS3 for which program usage data is to be collected.

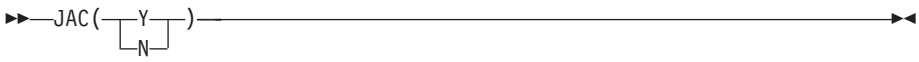
```
XDS(SYS3.*)
IDS(SYS3.LINKLIB)
```

JAC - Set job account collection status

JAC is used to specify if the Usage Monitor is to consider the account code of jobs significant when aggregating data. The Usage Monitor normally aggregates data based on the program name, the job name, and the user ID. This setting is used to add the job account, truncated after 20 characters, to the aggregation key.

Do not instruct the Usage Monitor to collect and preserve all job account codes if they are not important to the administration of your system. Collecting and preserving job accounts significantly increases data volumes.

A change to this setting does not take effect until the next data space repository switch.



Y Specifies that job account codes are used.

N Specifies that job account codes are ignored.

If no JAC command is issued after IPL, then job accounts are not used. The default is JAC(N).

JNM - Control the collection of job names

JNM is used to specify whether the Usage Monitor collects the names of jobs which use programs or not. If the names of jobs which use the various programs are not considered to be important, you can dispense with the collection of these names. The advantage of not collecting individual job names is the reduction in processing times and data volumes caused by the aggregation of data into fewer records. When individual job names are not collected, usage is summed over broad address space categories, such as JOB, STC, TSO, and SYS. The total usage counts collected by the Usage Monitor for each program are not affected by this setting.

A change to this setting takes effect at the next data space repository switch.

Y Specifies that the name of each job running a program is to be collected.

N Specifies that only a broad address space category of each job running a program is to be collected, instead of the individual job name.

If no JNM command has been issued since IPL, then job names are collected. JNM(Y) is the default.

LLC - Link list correction

LLC is used where sites make a number of dynamic link list changes. This command updates the HSIJMON data to point to the correct load library. Use this command only if you enable dynamic link list updates, which alter the relative concatenation numbers of persisting libraries.

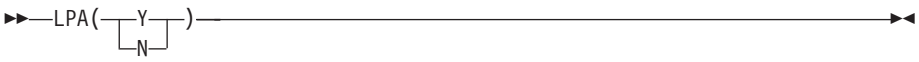
Y A BLDL is performed at write time by the writer task and, if found, the data set name is overlaid.

N Do not check for dynamic list updates.

If no LLC command is issued after IPL, then the default setting of LLC(N) is current.

LPA - Set link pack area program monitoring status

LPA is used to specify whether the monitoring of programs in the Link Pack Area (LPA) is to occur or not. All types of LPA are included in this category.



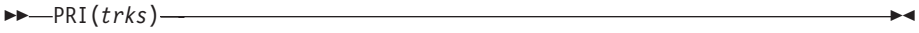
Y Specifies that LPA program usage is to be monitored.

N Specifies that LPA program usage is not to be monitored.

If no LPA command is issued after IPL, then LPA program usage data is collected. LPA(Y) is the default setting.

PRI - Set the data set space primary allocation

PRI is used to specify the primary space allocation quantity in tracks. It is used for output data set allocations.



trks Specifies a number of tracks from 0 to 150,000.

If no PRI command is issued, the primary space allocation is 750 tracks. The Usage Monitor uses the RLSE space allocation attribute.

Example

Set the primary space allocation to 900 tracks.

```
F HSIJMON,PRI(900)
```

PRS - Set registered software activity data collection status

PRS is used to specify if the Usage Monitor is to output records containing information about the activity of registered software. Registered software uses the system Register service. The data contains information about the usage of registered software, and information about software registration settings from member ISAPRDxx.

A change to this setting does not take effect until the next data space repository switch.

Y Specifies that registered software information is collected and output.

N Specifies that registered software information is neither collected or output.

If no PRS command is issued after IPL, then registered software data is collected. PRS(Y) is the default.

REF - Refresh Usage Monitor settings

REF is used at any time to reset Usage Monitor settings according to commands in the HSIZIN file, without stopping and starting the Usage Monitor. The detailed results of the refresh operation are written to the HSIZMSG file.

The processes of a refresh operation include:

- Verify that HSIZIN is still allocated.
- Open HSIZIN.
- Set the program exclusion list to the default list.
- Deactivate all data set exclusion list elements.

- Deactivate all data set inclusion list elements.
- Process the commands in HSIZIN.
- Close HSIZIN.
- Issue either HSIZ059I or HSIZ060I, as appropriate.

Example

Change Usage Monitor settings to updated values from HSIZIN.

```
F HSIJMON,REF
```

SEC - Set the data set space secondary allocation

SEC is used to specify the secondary space allocation quantity in tracks. It is used for output data set allocations.

trks Specifies a number of tracks from 0 to 150,000.

If no SEC command is issued, the secondary space allocation is 300 tracks. The Usage Monitor uses the RLSE space allocation attribute.

Example

Set the secondary space allocation to 600 tracks.

```
F HSIJMON,SEC(600)
```

SID - Set the Usage Monitor system identifier

SID is used to override the system identifier contained in the output header record. The SMF system identifier is used as a norm, but an override enables the data from separate systems to be differentiated in all instances where duplicate SMF identifiers are in use. Symbols can be employed in the construction of the system identifier. Available symbols include all z/OS system symbols, &SMF, the SMF identifier for the system, and &SYSLPAR, the logical partition name for the system.

sid Specifies a string which is to be resolved to an identifier 1-4 bytes in length.

Example 1

Set the output system identifier to PROD.

```
F HSIJMON,SID(PROD)
```

Example 2

Set the header record system identifier to the current LPAR name. The LPAR name must not exceed four characters in length.

```
F HSIJMON,SID(&SYSLPAR)
```

SIZ - Set the data space repository size

SIZ is used to specify the maximum number of entries that the data space repository can hold.

entries Specifies a number of entries from 100 to 6,000,000.

If no SIZ command is issued, a data space capacity of 200,000 entries is used. Each entry occupies 272 bytes. As each data space page has data placed in it for the first time, that page must be backed physically by the system. When a data space is full, a repository switch is triggered automatically. A repository switch also occurs when data stamped after the switch time is detected, or when a manual switch is registered by the **SWI** command.

Example

Set the size of future data spaces to 1,000,000 entries.

```
F HSIJMON,SIZ(1000000)
```

SWI - Switch to a new data space repository

SWI causes a new data space repository to be created and used for subsequent data collection. The data space being used at the time of SWI command is issued, has its data contents processed by a writer task.

The SWI command has no operands. It is invalid in the HSIZIN initial command file. As well as the switch caused by an explicit SWI command, automatic switches occur when a repository becomes full, and when data stamped after the switch time is detected. The SWI command might be rejected if the writer task is busy.

Example

Manually switch to a new repository.

```
F HSIJMON,SWI
```

SYM - Set the symbolic volume serial on output switch

SYM enables the logged volume serial number of a load library to be either the actual volume serial number or a symbolic value. When symbolic volume serial numbers are used, the IPL volume is always reported as six asterisks. For other volumes, if the serial number is found to match the value of a static system symbol, then the name of the symbol is reported instead of the actual volume serial number. Only symbols with names exactly six characters long are considered for this processing. The symbol name includes the leading ampersand (&), but excludes the trailing period (.).

Y The actual volume serial number is replaced by a symbolic value.

N The actual volume serial number is always output.

If no SYM command is issued, then the actual volume serial number collected is always output. SYM(N) is the default setting.

SYM(Y) is used with the SYMVOL operand of the Inquisitor.

TRG - Set the cache trigger event count

TRG enables the setting of the program usage event cache trigger.

Repository entries with usage counts greater than the trigger value are cached if there is space available. When usage events are captured for cached entries, usage of common storage and cross-memory POST processing is avoided.

When the cache is full, no additional entries can be cached. About every two hours, a status report, indicating cache usage, is written to the HSIZMSG file. The cache is then emptied, if at least half full. The cache is also emptied when the collection for a repository is stopped. A repository is stopped when the repository is switched, or when the Usage Monitor is shut down.

The maximum benefit of the cache occurs when the cache contains the entries which are collecting the most frequent program usage events.

The regular status reports in the HSIZMSG file should be examined to help determine the optimal cache trigger count which is often orders of magnitude larger than the default value.

count A number in the range from 0 to 999,999,999.

If no TRG command is issued after IPL, then the trigger count is set at 1000. TRG(0) specifies that the cache is to be filled as quickly as possible from the next captured program usage event data.

Example

Set the cache trigger event count to 1,200.

```
F HSIJMON,TRG(1200)
```

UID - Control the collection of user details

UID is used to specify whether the Usage Monitor collects the identifiers and names of users who use programs or not. If the details of users who use the various programs are not considered to be important, then you can dispense with the collection of this information. The advantage of not collecting user information is the reduction in processing times and data volumes.

When user information is not collected, the user ID data item is left blank, and user names are not output, regardless which UNM setting is current. The total usage counts collected by the Usage Monitor for each program are not affected by this setting.

If you want program usage attributed to individual users but do not want the names of users to be retained, use UID(Y) and UNM(N).

A change to this setting does not take effect until the next data space repository switch.

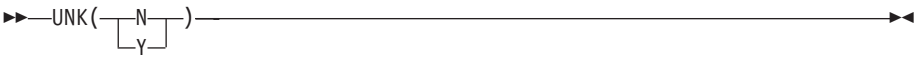
Y Specifies that details of each user using a program are to be collected.

N Specifies that details of each user using a program are **not** to be collected.

If no UID command is issued after IPL, user details are collected. UID(Y) is the default.

UNK - Set the unknown event collection switch

UNK is used to specify whether events with incomplete data are to be collected or not. The database content is not affected. Collecting extra data is useful in determining why some usage events are not captured. It must be set only when requested by IBM support.



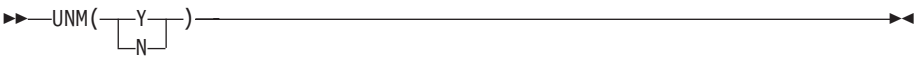
Y Specifies that the "unknown" events are to be collected.

N Specifies that the "unknown" events are not to be collected.

If no UNK command is issued after IPL, the unknown events are not collected. UNK(N) is the default setting.

UNM - Set user name collection status

Software security packages, such as RACF, have a name field for each user ID defined to the system. The Usage Monitor collects the user ID (up to eight characters long), and the contents of the name field (up to 20 characters long), as part of the data collection performed when programs are used. UNM is used to specify whether the names of users collected from the security package are output. The output of the user ID is controlled by the **UID** setting. This setting is checked by the writer task when the data in a data space repository is being processed for output.



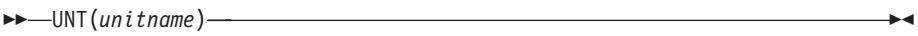
Y Specifies that collected user names are written to the output file.

N Specifies that collected user names are discarded.

If no UNM command is issued since after IPL, then user names are collected. UNM(Y) is the default.

UNT - Set the data set allocation unit

UNT is used to specify the allocation unit to be used for output data set allocations.



Unitname

Specifies a 1 - 8 character long unit name.

If no UNT command is issued, SYSALLDA is used.

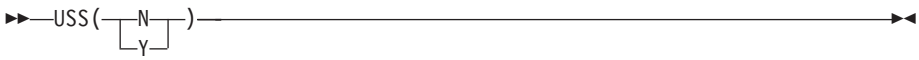
Example

Set the allocation unit to WORKDA.

```
F HSIJMON,UNT(WORKDA)
```

USS - Set UNIX program monitoring status

USS is used to determine if the programs retrieved from Hierarchical File System (HFS) files are to be monitored.



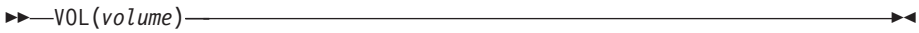
Y Programs fetched from HFS files are to be monitored.

N Programs fetched from HFS files are not to be monitored.

If no USS command is issued after IPL, the programs retrieved from HFS files are not monitored. USS(N) is the default setting.

VOL - Set the data set allocation volume

VOL is used to specify the allocation volume to be used for output data set allocations. The explicit nomination of a specific volume is necessary when there are no PUBLIC or STORAGE volumes in the allocation unit pool.



volume specifies a 1 - 6 character long volume serial number.

If no VOL command is issued, a specific volume is not explicitly requested. You must then have PUBLIC or STORAGE volumes in the public allocation pool, unless the data sets are managed by SMS.

Example

Set the allocation volume to SCR001.

```
F HSIJMON,VOL(SCR001)
```

WRT- Set the automatic switch-and-write time of day

WRT is used to specify a time-of-day to end data collection for the current data space, and automatically switch to a new data space. The data write-out for the closed data space is also initiated. These events are triggered when data from after the specified time is detected.

The UTC or GMT switch time is calculated using the local time when the data space is created. The time that a data space is terminated is set when it is created. Changes to the system local time offset, such as those caused by a change to daylight saving time status, do not alter the UTC or GMT time that the current data space is closed. The time of the switch, after the next switch, is calculated using the new local time.

hhmm Specifies a 24-hour time-of-day in hour and minute notation. The value must be four decimal digits. The first two digits (hh) must be in the 00 - 23 range. The last two digits (mm) must be in the 00 - 59 range.

If no WRT command is issued, the automatic switch time of midnight is used. That is, WRT(0000) is the default.

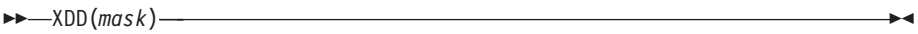
Example

Set the automatic switch-and-write time to 10 minutes before midnight.

```
F HSIJMON,WRT(2350)
```

XDD - Deleting data set name exclusion entries

XDD is used to remove data set name masks which were added by the XDS command.



mask Specifies a 1 - 44 character data set name mask. Any wildcard characters in the mask are treated as literals for the purposes of finding the mask to delete.

Example

Deactivate the SYS3.* exclusion mask.

```
F HSIJMON,XDD(SYS3.*)
```

XDS - Adding data set name exclusion entries

XDS is used to supply data set name masks which specify data set names to be excluded from data collection. Program usage data for programs fetched from data sets with names matching exclusion masks is discarded. When the captured data set name has been matched to an inclusion mask set by the IDS command, the data is collected without reference to the exclusion mask list.

mask Specifies a 1 - 44 character data set name mask. If the mask ends in an asterisk, only characters before the asterisk are compared. Percent signs in the mask indicate that any character in that location is considered a match.

Example

Exclude program usage data from collection for programs fetched from data sets with a high-level qualifier of SYS3.

```
F HSIJMON,XDS(SYS3.*)
```

ZIP - Set the compressed output data switch

ZIP is used to control whether the writer task is to compress output data or not. Compressing the output data reduces data volume, in turn reducing data transfer time and storage space requirements.

Y Specifies that output data is to be compressed.

N Specifies that output data is not to be compressed.

If no ZIP command is issued, then compressed data is output. ZIP(Y) is the default setting.

Inquisitor Import

The Inquisitor Import is the process of loading Inquisitor data into the Repository tables and does the following:

- Imports Inquisitor data generated from Inquisitor scans into Inquisitor tables. To exclude importing specific libraries, the Inquisitor data is filtered against a set of supplied Inquisitor Filter tables. These Inquisitor Filter tables are updated monthly, together with the Knowledge databases. The filtering excludes, for example, the ISV distribution libraries.

- Matches load modules in the Inquisitor tables to best fitting products at the Version Release Modification (VRM) level. Best matches for modules are found based on module names and sizes, and information in the Global Knowledge Base (GKB) and Local Knowledge Base (LKB). Temporary scorecard tables are used to hold all the possible scorecards for modules in a given library while they are matched. At the end of the matching step, data in the Inquisitor tables are updated with matched product information.
- Copies the Inquisitor data, including matching information, from the Inquisitor tables into the Repository tables. Data from the Repository tables are now ready for viewing or reporting using the Analyzer reporting facility.
- Aggregates usage data for rediscovered modules in the Repository tables.

Note: The Match Engine programs uses DB2 temporary work files as intermediate work areas and, by default, DB2 Global Temporary Tables are used. This means that temporary tables and indexes are created and dropped after each Match Engine session. Note: In DB2 Version 8, `Declare Global Temporary Table` requires at least one 8K table space to be defined in a TEMP database.

Running the Inquisitor Import

To run the Inquisitor Import, use the job HSISIQIM, in the JCLLIB. This job is generated from the HSISCUST post-installation customization job. Run time for this job depends on the number of modules to be imported into the DB2 Inquisitor tables. It is recommended to run HSISIQIM during off-peak periods.

When running job HSISIQIM, two parameters, FULLREMATCH and PRODUCTONLY, will influence the duration of the run. These two parameters are provided explicitly in the HSISIQIM job for you to update according to your requirements. See the description of these parameters below. Other parameters, which are less likely to be modified, are in a PARMLIB member.

TPARAM parameters

COMMIT=
Default is 1000. Number of records stored before issuing a COMMIT.

DSN= DB2 location. Value assigned, as defined in job HSISCUST

FILTERSCHEMA=
Inquisitor Import filter qualifier. Name of qualifier is *&DBGKB*_IQF7

FULLREMATCH=
Default is N, which means import and match processing will be skipped for scanned libraries that have had no member directory changes since a previous Inquisitor Import and into the same Repository. Y means that all libraries will be imported and matched.

GKBSCHEMA=
Global Knowledge Base qualifier for z/OS. Name of qualifier is *&DBGKB*_GKB7

GKUSCHEMA=
Global Knowledge Base qualifier for z/OS UNIX. Name of qualifier is *&DBGKB*_GKU7

IQSCHEMA=
Inquisitor qualifier for z/OS. Name of qualifier is *&DB*_ZIQ

IQUSCHEMA=
Inquisitor qualifier for z/OS UNIX. Name of qualifier is *&DB*_UIQ

LKBSCHEMA=
Local Knowledge Base qualifier for z/OS. Name of qualifier is *&DB*_LKB7

LKUSCHEMA=
Local Knowledge Base qualifier for z/OS UNIX. Name of qualifier is *&DB*_LKU7

PRODUCTONLY=
Default is N, which means all modules, including unidentified modules, are loaded into the Repository. Y means only modules that have been matched to known products are loaded into the Repository, meaning, application modules are excluded.

REPSCHEMA=
Repository qualifier. Name of qualifier is *&DB*.

Usage Import

Usage Import is the process of importing usage data into the Repository tables. Usage Import performs these tasks:

- Imports usage data generated from the Usage Monitor.
- Aggregates usage data for discovered or identified modules in the Repository tables.

Running Usage Import

To run the Usage Import, use the job HSISUIMP in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

Run time for this job depends on the amount of usage data to be loaded into the DB2 repository tables. It is recommended to run this job during off-peak periods.

TPARAM parameters

COMMIT=
Default is 1000. Number of records stored before issuing a COMMIT.

DSN= DB2 location. Value assigned, as defined in job HSISCUST.

REPSCHEMA=
Repository qualifier. Name of qualifier is *&DB*.

Automation Server for z/OS

The Automation Server is a utility that automates data set processing. It discovers newly created data sets and processes them by starting a set of predefined actions. Each action definition is associated with one or more data set name masks that are used as arguments in a catalog search. This search determines if any data set names matching the mask are to be processed. The ability to select data sets is useful if you have data set names that are variables, such as those created by the Usage Monitor, which have low-level qualifiers containing time stamps.

The Automation Server runs in its own address space as a started task. You must ensure that the user ID employed by the Automation Server has an OMVS segment and UID, or use of a default UID.

Input control statements define the processing to be performed by the Automation Server. There are two types of control statements, *action* statements and DSN statements:

action Action statements name the template member which forms the basic input for the action to be performed when a relevant data set is newly discovered by a catalog search. They have optional operands to specify time-of-day, day-of-week, and day-of-month restrictions.

DSN DSN statements provide a data set name mask to be associated with the preceding action statement. There can be many DSN statements after each action statement.

There are currently two types of action statement:

FTP Starts the FTP utility to perform a file transfer.

For the FTP action, the template member is read and, after symbol substitution processing, is written to the file defined by the INPUT DD statement. The FTP program is attached as a subtask and scans the INPUT file to process the FTP requests. This INPUT file is allocated to a temporary data set. The report messages it generates are written to the OUTPUT file.

Upon completion of the FTP subtask, the Automation Server examines the completion code. If the program ends normally with a zero return code, the Automation Server deems the action to have been successful and updates the action status in the HSIACDS file so the action is not repeated for this data set.

If the FTP program abends, the Automation Server deems the action to have failed. A failed transfer is tried again at a later time. A retry is subject to specified time-of-day window constraints. The OUTPUT FTP report file contains information to track the exact cause of a transfer failure.

JOB Submits a batch job.

For the JOB action, the template member is read and, after symbol substitution processing, is written to the file defined by the INTRDR DD statement. This file is directed to the internal reader used by the system, and the jobs submitted by the Automation Server become available for JCL conversion as soon as the INTRDR file is closed, or another JOB card image is found by the reader.

The Automation Server deems all JOB submissions successful, so there are no retries. Any failure should be investigated using the appropriate procedures used by your installation.

Note: The job stream in a JOB action template member may define more than one job.

The Automation Server does not check template member records for either FTP or JCL validity.

Running the Automation Server

This section details the following tasks:

- "1. Creating the Automation Server control data set" on page 68.
- "2. Copying the started task JCL to a library" on page 68.
- "3. Designing request control data sets" on page 69.

- "4. Starting and stopping the Automation Server" on page 73.
- "5. Excluding data sets from Automation Server processing" on page 73.

1. Creating the Automation Server control data set

To create the Automation Server control data set, use member HSIASALC in the JCLLIB. This member is generated from the HSISCUST post-installation customization job.

Member HSIASALC contains IDCAMS JCL and control statements to create the control data set.

The Automation Server control data set is a VSAM KSDS that is allocated by the Automation Server with the HSIACDS ddname.

Note: The Automation Server control data set must have sufficient space allocated for it to handle the workload required by the installation. One 96 byte record (including the 52 byte key) is required for each data set processed by the Automation Server.

2. Copying the started task JCL to a library

The Automation Server started task JCL **HSIJAUTO** is supplied in the SHSIPROC data set.

To start HSIJAUTO as a Started Task, copy the customized member from the JCLLIB to an authorized PROCLIB.

Use the parameters listed in this table in the started task JCL:

Table 20. Started task JCL parameters

Statement Type	Description
HSI	High-level qualifiers for the Installation Target Libraries months
HSIINST	High-level qualifiers for the &HSIINST..PARMLIB data set created by the HSISCUST job.
ACDS	Data set name of the Automation Control Data Set (ACDS).

Files used by the Automation Server:

STEPLIB
Load library containing the product software. Not required if Tivoli Asset Discovery for z/OS is installed into the system link list.

HSIACNTL
Partitioned data set containing fixed length 80 byte records. Member HSIAPARM of this partitioned data set contains the Automation Server control statements that specify the actions to be performed. For each action in the HSIAPARM member, there is a corresponding member of the same name containing the template data for that action. The template data is made up of JCL or an FTP command stream containing symbolic references, to be resolved by the Automation Server when the action is performed.

HSIACDS
A VSAM KSDS control data set used by the Automation Server.

HSIAMSG
Specifies the message report file for the Automation Server. Initialization statements, error messages, and activity logging messages, are written to this file.

SYSPRINT
Specifies the message report file for Language Environment.

SYSOUT
Specifies the message report file for Language Environment.

OUTPUT
Specifies the message report file for the FTP program. The contents are determined by the FTP program installed in the system.

INPUT
Specifies a fixed length 120 byte record file containing FTP commands read by the FTP program. The FTP commands are written to this file before the Automation Server FTP action is performed.

INTRDR
Specifies a fixed length 80 byte record file to be directed to the internal reader used by the system. The Automation Server writes a job stream to this file whenever a JOB action is to be performed.

3. Designing request control data sets

The Automation Server action requests are specified in the HSIAPARM member of the SHSIPARM file.

Syntax rules are as follows:

- Records with an asterisk in column 1 are comments.
- Blank records are comments.
- A parameter record has one or more parameters, each with a value specified within parentheses after the parameter name.
- The first parameter specifies the statement type.
- All parameters must begin before column 72.
- Blanks can be used before and after parameter names, parentheses, and parameter values.
- Continuations on to subsequent records are not possible.

Action statement
Each statement requests that an action is performed for a data set when it matches an associated data set name mask, and is detected for the first time. An action is performed once for each match, but the presence of a data set triggers the action for each specified data set name mask it matches.

Action statements have several optional operands to provide control over when Automation Server processing is to occur.

These operands can specify:

- time-of-day window
- day-of-week control string
- day-of-month window

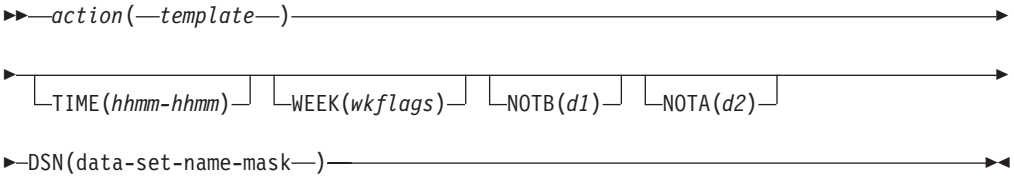
When all these constraints have been satisfied, the Automation Server searches the catalog for data sets with names that match the masks associated with an action.

Data set name mask statement

Each data set name mask statement associates the specified data set name mask with the preceding action statement. It is invalid for the HSIAPARM member to begin with a data set name mask statement. When a data set with a name matching the specified mask is first located, the action specified in the preceding action statement is triggered.

The data set name mask of NULLFILE is an exception. When a data set name mask with this exact value is processed by the Automation Server, a catalog search is not performed, but the associated action is triggered as if a new cataloged data set matching the mask has been located. Automation Server symbols for the data set name, and symbols for the first qualifier of the data set name, both have values of the 8 byte string NULLFILE. Use the data set name mask of NULLFILE to trigger scheduled actions which do not depend on the creation of a particular data set.

Action statement and DSN syntax



action FTP or JOB

template

Name of a member in the HSICNTL file.

TIME This operand is optional, and the default is TIME(0000-2400), which specifies no time-of-day constraint.

hhmm-hhmm

Specifies a time-of-day range. Each *hhmm* value is four contiguous decimal digits that specify a time-of-day using the 24 hour clock. The minimum value is 0000 and the maximum value is 2400; the last two digits must not exceed 59. The two values are separated by a hyphen. Zero or additional blanks are also permitted. The first *hhmm* specifies the time-of-day window start, while the second specifies the time-of-day window end. The window includes times which are after the window start and before the window end. However, if the second *hhmm* value is lower than the first, the window includes times which are after the window start or before the window end.

WEEK This operand is optional. The default is WEEK(YYYYYYY), which specifies that the *action* can be run on every day of the week.

wkflags

Specifies a single contiguous 7 byte string, consisting of the uppercase characters Y or N. Each Y or N corresponds to a day of the week depending on its position in the string; the first corresponding to Sunday, the last to Saturday. If the character corresponding to a day of the week is N, the action is not processed on that day.

NOTB This operand is optional. The default NOTB(1) specifies that the monthly window starts on the first possible day of the month. NOTB means "not before".

d1 Specifies a one or two digit decimal number in the 1-31 range. This number denotes the first possible day of the month on which the action is permitted.

NOTA This operand is optional. The default NOTA(31) specifies that the monthly window extends to the last day of the month. NOTA means "not after".

d2 Specifies a one or two digit decimal number in the 1-31 range. This number denotes the last possible day of the month on which the action is permitted.

DSN Data set name.

data-set-name-mask
Specifies a data set name mask pattern which does not exceed 44 characters in length, and is used by the Catalog Search Interface. The generic match mask for a single character is the percent sign. The generic match mask variable number of characters is the asterisk. A double asterisk can be used to match a variable number of data set name qualifiers.

Control statement examples

Example 1:
Files created by the Usage Monitor undergo two independent processes, both within the 8:00 p.m. to 11:30 p.m. window. They are processed by a job based on the JCL contained in member HSISJOB, and are separately transferred to a z/OS system using the FTP commands in member HSISFTP1. All members are pointed to by the HSIACNTL ddname.

```
* TRANSFER USAGE MONITOR FILES TO Z/OS SYSTEM
   JOB(HSISJOB1)  TIME(2000-2330)
   DSN(USER.OMU*.D*.T*)
   FTP(HSISFTP1)  TIME(2000-2330)
   DSN(USER.OMU*.D*.T*)
```

Example 2:
Files created by the Usage Monitor are to be imported to the appropriate database.

```
* PERFORM USAGE MONITOR IMPORT
JOB(HSISUIMP)
DSN(USER.UMON.*.*)
```

In this example HSISUIMP contains the necessary JCL to run Usage Import on a z/OS system.

Note: The JCL can route the job to any connected NJE node, or specify an affinity to any system sharing the SPOOL. You do not need to run the job on the z/OS system where the Automation Server is running. The template name, HSISUIMP in this example, does not need to match the job name submitted by the Automation Server action.

Example 3:
A job stream stored in member WED2MNTH is to be submitted unconditionally on the second Wednesday of every month.

```
* RUN MONTHLY JOBSTREAM ON THE SECOND WEDNESDAY OF EVERY MONTH
 JOB ( WED2MNTH )   WEEK ( NNNYNNN )   NOTB ( 8 )   NOTA ( 14 )
 DSN ( NULLFILE )
```

Automation Server symbol processing:

Whenever an action is performed, the contents of the template member are written to an appropriate output file. Each 80 byte record is written unchanged, unless symbol substitution is required. If an ampersand character is present in a record from the template member, the system symbol substitution routine ASASYMBM is started, which processes the record before it is written. You can use more than one symbol in a record. If an ampersand character does not denote the start of a recognized symbol, then that part of the data remains unchanged. Symbols available for use in template members include all z/OS system symbols and symbols defined locally by the Automation Server. Most Automation Server local symbols are derived from the catalog entry data set name which, when discovered, triggers the instance of the action.

System symbols supplied by the operating system, as well as the &SMF and &SYSLPAR symbols supplied by the Automation Server, are available for use in the HSIAPARM member. The &SYSLPAR symbol might resolve to a null string if the system is running in a virtual machine.

Automation Server local symbols are provided in the following table:

Table 21. Automation Server local symbols

&SMF	System SMF identifier.
&SYSLPAR	System LPAR name.
&DATASETNAME.	The entire data set name.
&QUAL1.	The first qualifier of the data set name.
&QUAL2.	The second qualifier of the data set name.
&QUAL3.	The third qualifier of the data set name.
&QUAL4.	The fourth qualifier of the data set name.
&QUAL5.	The fifth qualifier of the data set name.
&QUAL6.	The sixth qualifier of the data set name.
&QUAL7.	The seventh qualifier of the data set name.
&QUAL8.	The eighth qualifier of the data set name.
&QUAL9.	The ninth qualifier of the data set name.

Automation Server symbol processing example

Example:

The data set triggering a JOB action is IBMUSER.IQ.ZIP. As a result, JCL DD statements referencing the data set in a template member can be represented as shown in this example:

```
//*-------------------------------------------------------------***
//* Sample JCL demonstrating the use of Automation Server local***
//* symbols derived from the data set name.                     ***
//*-------------------------------------------------------------***
//BR14      EXEC  PGM=IEFBR14
//DD1       DD    DSN=&&DATASETNAME.,DISP=SHR
//DD2       DD    DSN=&&QUAL1..&&QUAL2..&&QUAL3.,DISP=SHR
```

Both JCL DD statements would be resolved by symbol substitution to:

```
DSN=IBMUSER.IQ.ZIP,DISP=SHR
```

This is the **DSN=** JCL statement output to the internal reader.

Note: As symbol substitution is performed before the job is submitted, z/OS system symbols that cannot be used in batch job JCL, can be used in the Automation Server templates.

4. Starting and stopping the Automation Server

Starting

When installed as a started task, the Automation Server must be started by an operator **START** command. Ensure that the user ID assigned to the Automation Server has RACF CONTROL access to the VSAM data set.

Stopping

When running as a started task or as a batch job, the Automation Server is stopped by an **MVS STOP** command.

5. Excluding data sets from Automation Server processing

In the HSIAPARM member you define actions to be performed, and supply data set name masks specifying the data sets to be processed. Data sets with these name patterns might already exist and been processed before the Automation Server was implemented. Data sets that you want excluded from the Automation Server processing must have a record in the Automation Server control data set indicating that the data set has already been processed. The name of the data set you want to exclude must satisfy a selection mask pattern. To implement this exclusion, you can use the Automation Server data set name scouting program. The program reads the HSIAPARM member and searches the catalog for every specified data set name mask. If a data set is found, a record is written, then sorted into key order, and copied into the VSAM control data set. Every record loaded into the control data set in this way indicates a specific action, and has that action flagged as complete.

You can edit the sequential data set before the data is copied into the control data set. This process manually deletes any records for data sets you still want processed by the Automation Server.

To run the scouting program, the HSISCUST post-installation customization job creates member HSIASSCT in the JCLLIB.

Automation Server control data set maintenance

A record is kept for every data set processed by the Automation Server in the Automation Server control data set, ACDS. The purpose of this record is to prevent the repeated processing of a data set for the same data set name mask. As records accrue, the size of the data in the ACDS continues to grow.

If a processed data set is deleted, or a data set name mask is removed from the set of masks processed by the Automation Server, then there is no reason to keep a record of that data set in the ACDS. The Automation Server performs a cleanup cycle for the ACDS on a daily basis. This cleanup cycle consists of reading the ACDS sequentially, and deleting records for data sets which have not been found by catalog search. This is based on the relevant data set name mask in the current calendar month, or in the prior calendar month.

As with most VSAM data sets with ongoing record insertion and deletion activity, it is advisable to periodically reorganize the ACDS.

Chapter 5. Reporting

The primary reporting facility in Tivoli Asset Discovery for z/OS is the Analyzer. In addition, you can optionally use Tivoli Common Reporting to develop your own custom reports.

The Analyzer

The Analyzer runs as a started task or batch job on the same central z/OS host as the DB2 Subsystem that contains the Tivoli Asset Discovery for z/OS database(s).

The Analyzer has two modes:

Online mode
A PC Browser, for example Firefox, is used to communicate with the Analyzer for interactive queries.

Batch mode
This mode uses the Analyzer to generate the report to an output file. The Batch mode is useful when you want to automate report your own reports.

All Analyzer reports can be run in online and batch modes and can produce the following output formats:

- HTML (htm)
- Excel (Excel)
- Text line (txt)
- Comma Separated Value (csv)

This section explains:

- Analyzer - Getting Started
- Analyzer Pre-Requisites
- Analyzer JCLLIB and PARMLIB members
- Analyzer National Language Support
- Running Analyzer in online mode
- Running Analyzer in batch mode
- Controlling the Analyzer Address Space
- Analyzer Custom Queries

See Appendix B, “Analyzer,” on page 169 for information on:

- Analyzer online mode navigation
- Analyzer reports
- Analyzer report parameters
- Analyzer report output columns

Getting started with the Analyzer

The Analyzer has several configuration options. In order to familiarize yourself with the Analyzer, it is a good idea to first run the Analyzer with the default settings. To get started, follow these steps:

1. Run the HSISANLO batch job, which is found in the JCLLIB, on the same z/OS host as the DB2 Subsystem that contains the Tivoli Asset Discovery for z/OS database(s). The person who submits the HSISANLO batch job must have already been granted access to the databases by using the HSISGRNT job in the JCLLIB. By default, HSISANLO runs the Analyzer in online mode with basic security and http port 9000.
2. Use your PC web browser to logon with the URL http://*hostname*:9000, where*hostname* is your z/OS IP host name or IP address. If you are not sure of the name, look at the Analyzer log, where the URL is listed.
3. At the login screen specify a user ID of `TADZADM` and a password of `TADZ`.

For more information, see Appendix B, "Analyzer," on page 169.

Analyzer prerequisites

The Analyzer uses the DB2 Call Library Interface, also used by the Usage Import component, and the z/OS socket application programming interface.

There is no dependency on any other middleware components. For example, no dependency exists on the HTTP Server, WebSphere® Application Server, or Java.

The Analyzer has been designed with minimal prerequisites. These are:

- The Analyzer must be run on the same z/OS host as the DB2 Subsystem that contains the Tivoli Asset Discovery for z/OS databases.
- The user ID of the Analyzer address space must have previously been granted access to the databases. See the HSISGRNT job in the JCLLIB for sample JCL to grant access.
- When running the Analyzer in the online mode, you need access to a TCP/IP port. The default is port 9000.
- When running the Analyzer in online mode with SECURITY=SYSTEM, the Analyzer SHSIMOD1 load library must be defined to the z/OS Authorized Program Facility (APF). In addition, all data sets in the Analyzer STEPLIB, or JOBLIB DD concatenation, must be defined to APF.

Analyzer JCLLIB and PARMLIB members

The members in the JCLLIB contain sample JCL to run the Analyzer.

Table 22. JCLLIB members for Analyzer

Member	Description
HSIJANLO	Analyzer PROC for online mode. Copy this PROC from the JCLLIB to a system PROCLIB data set to run the Analyzer as a started task
HSISANLB	Analyzer batch job for batch mode
HSISANLO	Analyzer batch job for online mode
HSISANS1	Define the Analyzer security profiles in RACF (only applicable for Analyzer SECURITY=SYSTEM setting)
HSISANS2	Generate the Analyzer SSL certificate in RACF (only applicable for Analyzer SECURITY=SYSTEM setting)
HSISANS3	Connect the Analyzer user ID to an existing SSL certificate in RACF (only applicable for Analyzer SECURITY=SYSTEM setting)

The following members in the PARMLIB contain sample configuration settings for the Analyzer in online mode. These members are referenced with the TPARAM setting in the HSIJANLO PROC.

Table 23. PARMLIB members for Analyzer

Member	Description
HSISANP1	Basic Security: • Security settings as defined in the Analyzer TPARAM DD • HTTPS (SSL encrypted) communications
HSISANP2	Analyzer batch job for batch mode

Analyzer globalization support

By default, the Analyzer uses English for all report titles, headings, and descriptions. To change to Japanese, define the HSINLS DD to point to the HSINLSJP member in the SHSIANL1 data set.

You can also define your own custom language settings with the HSINLS DD. HSINLSEN member in SHSIANL1 data set provides a template. It contains English key phrases on the left of the screen assigned to text that is used on the reports.

Running the Analyzer in online mode

To run the Analyzer in online mode as a Started Task, copy the HSIJANLO from the JCLLIB to a system PROCLIB data set.

```
//HSIJANLO PROC HSI='TADZ.V750',       TADz Target library HLQ.
//            HSISCLI='HSISCLI',       DB2 CLI Parms
//            TPARAM='HSISANP1'        TPARAM input parms
//*
//ANALYZER EXEC PGM=HSICANLZ,REGION=0M,TIME=NOLIMIT
//STEPLIB  DD DISP=SHR,DSN=&HSI..SHSIMOD1
//         DD DISP=SHR,DSN=&DB2EXIT
//         DD DISP=SHR,DSN=&DB2LOAD
//SYSPRINT DD SYSOUT=*,HOLD=YES,LRECL=500
//HSIANL1  DD DISP=SHR,DSN=&HSI..SHSIANL1
//HSIANL2  DD DISP=SHR,DSN=&HSI..SHSIANL2
//DSNAOINI DD DISP=SHR,DSN=&HSIINST..PARMLIB(&HSISCLI)
//HSICUST  DD DISP=SHR,DSN=&HSIINST..PARMLIB(HSISANCQ)
//*HSINLS   DD DISP=SHR,DSN=&HSI..SHSIANL1(HSINLSJP)
//TPARAM   DD DISP=SHR,DSN=&HSIINST..PARMLIB(&TPARAM)
```

HSISANLO job in the JCLLIB is used to run Analyzer in online mode as a batch job.

```
//HSISANLO EXEC HSIJANLO,TPARAM=HSISANP1
```

When the Analyzer is run with online mode, configuration options must be defined in the TPARAM DD, including the communication port and security mode.

Analyzer communication port

The Analyzer communication port is defined by using the HTTPPORT setting. Both sample PARMLIB members HSISANP1 (basic security), and HSISANP2 (system security), have the following:

```
*********************************************************************
* HTTPPORT defines the TCP/IP port used for communications.          *
*                                                                   *
*   If HTTPPORT = 9000  is defined on a system with a TCP/IP host    *
*   called sys1.mycompany.com, to access the TADz Analyzer the user  *
*   would specify the following URL in their PC Browser:            *
*        http://sys1.mycompany.com:9000  if SECURITY=BASIC          *
*    or https://sys1.mycompany.com:9000  if SECURITY=SYSTEM         *
*                                                                   *
*   The port specified must be available on your system.            *
*                                                                   *
*   TSO NETSTAT can be used to check if a port is available e.g.:    *
*   TSO NETSTAT (PORT 9000       --*  is port 9000 in use?         *
*   TSO NETSTAT PORTL(PORT 9000  --*  is port 9000 reserved?       *
*                                                                   *
*   If no entries are returned from these NETSTAT commands, the port *
*   is most probably available.  At some sites, you may need your    *
*   Network Systems Programmer to reserve a port for TADz Analyzer.  *
*                                                                   *
*********************************************************************
HTTPPORT = 9000
```

If HTTPPORT is not specified, or is set to 0, the Analyzer runs in batch mode instead of in online mode

Analyzer BASIC security

HSISANP1 in the PARMLIB defines the following basic user ID security settings.

Note: User IDs TADZADM and TADZUSR can be used without any prior configuration. User ID AUID001 is a sample of how to restrict a user ID to certain databases.

```
*********************************************************************
* SECURITY=BASIC  - HTTP communications                             *
*                     with basic security defined in TPARAM DD below. *
*                                                                   *
*********************************************************************
 SECURITY = BASIC

*********************************************************************
* The following settings are only applicable for                    *
* SECURITY=BASIC:                                                   *
*                                                                   *
*   AUTH_USER defines Userids and passwords for Analyzer logon      *
*   AUTH_DB   defines the databases access                          *
*   AUTH_MENU defines the menus access                              *
*                                                                   *
* The sample settings profile:                                      *
*  - TADZADM userid:                                                *
*       - Password TADZ                                             *
*       - Access to all databases                                   *
*       - Access to all menu tabs                                   *
*  - TADZUSR userid:                                                *
*       - Password TADZ                                             *
*       - Access to all databases                                   *
*       - Access to menu tabs ASSET, DISC + CUSTOM only (not ADMIN) *
*  - AUID001 userid:                                                *
*       - Password PW01                                             *
*       - Access to databases AUDB01 + AUDB02 only                  *
*       - Access to menu tab ASSET only                             *
*********************************************************************

*-------------------------------------------------------------------
*AUTH_USER= USERID   , PASSWORD                                     *
*-------------------------------------------------------------------
```

```
AUTH_USER = TADZADM  , TADZ
AUTH_USER = TADZUSR  , TADZ
AUTH_USER = AUID001  , PW01

*-------------------------------------------------------------------*
*AUTH_DB  = DATABASE , LIST OF USERIDS AUTHORIZED TO SEE THE DATABASE *
*-------------------------------------------------------------------*
AUTH_DB   = *        , TADZADM TADZUSR
AUTH_DB   = AUDB01   , AUID001
AUTH_DB   = AUDB02   , AUID001

*-------------------------------------------------------------------*
*AUTH_MENU= MENU_TAB , LIST IF USERIDS AUTHORIZED TO SEE THE MENU TAB *
*-------------------------------------------------------------------*
AUTH_MENU = ASSET    , TADZADM TADZUSR AUID001
AUTH_MENU = DISC     , TADZADM TADZUSR
AUTH_MENU = ADMIN    , TADZADM
AUTH_MENU = CUSTOM   , TADZADM TADZUSR
```

Analyzer SYSTEM security

HSISANP2 in the PARMLIB defines the following system security settings:

```
*********************************************************************
* SECURITY=SYSTEM - HTTPS (SSL encrypted) communications             *
*                    with z/OS system security (SAF/RACF).          *
*                    Refer to HSISANS1/2/3 in JCLLIB for sample JCL  *
*                    to define RACF profiles/certificates.          *
*                                                                   *
*********************************************************************
SECURITY = SYSTEM

*********************************************************************
* The following settings are only applicable for                    *
* SECURITY=SYSTEM:                                                  *
*                                                                   *
*   AUTH_HLQ         defines SAF/RACF profile high level qualifier   *
*                                                                   *
*   AUTH_UPPERCASE=Y Analyzer will uppercase passwords when          *
*                    invoking SAF/RACF password authentication      *
*   AUTH_UPPERCASE=N Analyzer will pass through mixed case passwords *
*                    when invoking SAF/RACF password authentication *
*                                                                   *
*   GSK_KEYRING_FILE defines SAF/RACF Keyring name of SSL Certificate *
*   GSK_KEY_LABEL    defines SAF/RACF Label   name of SSL Certificate *
*   GSK_....         defines optional z/OS SSL environment variables. *
*                    The z/OS Cryptographic Cryptographic Services   *
*                    Secure Sockets Layer Programming manual         *
*                    SC24-5901-07 explains the environment variables. *
*                    For example, define GSK_HW_CRYPTO = 32          *
*                    for SHA-256 digest generation.                  *
*                                                                   *
* JCLLIB(HSISANS1) contains sample JCL to define RACF profiles, using *
* a high level qualifier of 'TADZ'.  If you have changed HSISANS1,   *
* you may also need to change the AUTH_HLQ TPARAM setting.          *
*                                                                   *
* JCLLIB(HSISANS2/3) contains sample JCL to define RACF SSL         *
* Certificates.  If you have changes HSISANS2/3, you may also need to *
* change the GSK_KEYRING_FILE and GSK_KEY_LABEL TPARAM settings.    *
*                                                                   *
*********************************************************************
AUTH_HLQ         = TADZ
AUTH_UPPERCASE   = Y
GSK_KEYRING_FILE = TADZ_KEYRING
GSK_KEY_LABEL    = TADZCERT
```

HSISANS1 in the JCLLIB has sample JCL to define RACF security profiles.

Note: The RACF ID can be an existing RACF group (which user IDs have been connected to) and/or existing RACF user IDs.

If your z/OS system has been set up to use a third party alternative to RACF, you must define comparable settings in your third party security product.

```
/*-------------------------------------------------------------*/
/* TADZ ANALYZER DATABASE PROFILES                             */
/*-------------------------------------------------------------*/
 RDELETE FACILITY  TADZ.DB.AU*
 RDEFINE FACILITY  TADZ.DB.AU*           UACC(NONE)
 PERMIT            TADZ.DB.AU*           ACCESS(READ) -
   CLASS(FACILITY) ID(TADZADM,TADZUSR,AUID001)

 RDELETE FACILITY  TADZ.DB.*
 RDEFINE FACILITY  TADZ.DB.*             UACC(NONE)
 PERMIT            TADZ.DB.*             ACCESS(READ) -
   CLASS(FACILITY) ID(TADZADM,TADZUSR)
 PERMIT            TADZ.DB.*             ACCESS(NONE) -
   CLASS(FACILITY) ID(AUID001)

/*-------------------------------------------------------------*/
/* TADZ ANALYZER MENU PROFILES                                 */
/*-------------------------------------------------------------*/
 RDELETE FACILITY  TADZ.MENU.ASSET
 RDEFINE FACILITY  TADZ.MENU.ASSET       UACC(NONE)
 PERMIT            TADZ.MENU.ASSET       ACCESS(READ) -
   CLASS(FACILITY) ID(TADZADM,TADZUSR,AUID001)

 RDELETE FACILITY  TADZ.MENU.DISC
 RDEFINE FACILITY  TADZ.MENU.DISC        UACC(NONE)
 PERMIT            TADZ.MENU.DISC        ACCESS(READ) -
   CLASS(FACILITY) ID(TADZADM,TADZUSR)

 RDELETE FACILITY  TADZ.MENU.ADMIN
 RDEFINE FACILITY  TADZ.MENU.ADMIN       UACC(NONE)
 PERMIT            TADZ.MENU.ADMIN       ACCESS(READ) -
   CLASS(FACILITY) ID(TADZADM)

 RDELETE FACILITY  TADZ.MENU.CUSTOM
 RDEFINE FACILITY  TADZ.MENU.CUSTOM      UACC(NONE)
 PERMIT            TADZ.MENU.CUSTOM      ACCESS(READ) -
   CLASS(FACILITY) ID(TADZADM,TADZUSR)

  SETROPTS RACLIST(FACILITY) REFRESH
```

SSL Certificates

When the Analyzer is running with SYSTEM=SECURITY, you must have an SSL Certificate defined in your SAF/RACF security system. You can either generate your own certificate, or connect to an existing certificate.

HSISANS2 in JCLLIB has sample JCL to generate SSL certificates in RACF.

```
//*********************************************************************
//*                                                                   *
//* To enable TADz Analyzer to use HTTP secure (HTTPS) the following  *
//* steps should be implemented by your site's RACF Administrator:    *
//* 1. Delete KEYRING(TADZ_KEYRING) and certificate with the          *
//*    LABEL('TADZCERT').                                             *
//* 2. Activate RACF Classes required for digital certificates.       *
//* 3. Define Keyring TADZ_KEYRING.                                   *
//* 4. Generate certificate.                                          *
//* 5. Connect to Keyring.                                            *
//* 6. Refresh RACF Classes required for digital certificates.        *
//* 7. Permit access to the Facility Class profiles.                  *
```

```
//*                                                                   *
//*                                                                   *
//* The following JCL demonstrates a sample implementation:           *
//* 1. Update all occurrences of "Userid-running-HSISANLO" to reflect *
//*    your TADz HTTPS environment.                                   *
//*                                                                   *
//* Do not change the RACF keyring 'TADZ_KEYRING' or label 'TADZCERT' *
//* unless you update the corresponding values in analyzer PARMLIB    *
//* member HSISANP2 and restart the Analyzer STC/Job.                 *
//*-------------------------------------------------------------------*
//RACFDEF    EXEC  PGM=IKJEFT01,DYNAMNBR=30
//SYSTSPRT   DD    SYSOUT=*
//SYSTSIN    DD    *
 PROF NOPREF

 RACDCERT DELETE(LABEL('TADZCERT'))
 RACDCERT ID(CMACN) DELRING(TADZ_KEYRING)

 SETROPTS CLASSACT(DIGTCERT,DIGTNMAP)
 SETROPTS RACLIST(DIGTCERT,DIGTNMAP)

 RACDCERT ID(Userid-running-HSISANLO) ADDRING(TADZ_KEYRING)

 RACDCERT ID(Userid-running-HSISANLO) CERTAUTH GENCERT -
 SUBJECTSDN( O('Your Organization')   -
 CN('Your Domain')                    -
 C('US')) TRUST                       -
 WITHLABEL('LOCALCA')                 -
 KEYUSAGE(CERTSIGN)

 RACDCERT ID(Userid-running-HSISANLO) GENCERT -
 SUBJECTSDN (CN('TADZCERT')                   -
 OU('Your Dept.')                             -
 C('US'))                                     -
 WITHLABEL('TADZCERT')                        -
 SIGNWITH(CERTAUTH                            -
 LABEL('LOCALCA'))

 RACDCERT ID(Userid-running-HSISANLO)         -
 CONNECT(ID(Userid-running-HSISANLO)          -
 LABEL('TADZCERT')                            -
 RING(TADZ_KEYRING)                           -
 DEFAULT                                      -
 USAGE(PERSONAL))

 RACDCERT ID(Userid-running-HSISANLO)         -
 CONNECT(ID(Userid-running-HSISANLO) CERTAUTH -
 LABEL('LOCALCA')                             -
 RING(TADZ_KEYRING)                           -
 USAGE(CERTAUTH))

 SETROPTS RACLIST(DIGTCERT,DIGTNMAP) REFRESH
/*
//PERMIT     EXEC  PGM=IKJEFT01,DYNAMNBR=30
//SYSTSPRT   DD    SYSOUT=*
//SYSTSIN    DD    *
 PROF NOPREF

  RDEL FACILITY IRR.DIGTCERT.LIST
  RDEL FACILITY IRR.DIGTCERT.LISTRING
  SETR RACLIST(FACILITY) REFRESH

  RDEFINE FACILITY IRR.DIGTCERT.LIST  UACC(NONE)
  RDEFINE FACILITY IRR.DIGTCERT.LISTRING  UACC(NONE)
  SETR RACLIST(FACILITY) REFRESH

  PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)     -
```

```
  ID(Userid-running-HSISANLO) AC(READ)

  PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) -
  ID(Userid-running-HSISANLO) AC(READ)

  SETR RACLIST(FACILITY) REFRESH
/*
```

HSISANS3 in JCLLIB has sample JCL to connect to existing SSL certificates in RACF.

```
//*********************************************************************
//*                                                                   *
//* To enable TADz Analyzer to use HTTP secure (HTTPS) using an       *
//* existing CA certificate, 'Entrust Secure Server Root CA' in our   *
//* example, the following steps should be implemented by your site's *
//* RACF Administrator:                                               *
//*                                                                   *
//* 1. Delete KEYRING(TADZ_KEYRING) and certificate with the          *
//*    LABEL('TADZCERT').                                             *
//* 2. Activate RACF Classes required for digital certificates.       *
//* 3. Define Keyring TADZ_KEYRING.                                   *
//* 4. Connect the existing CA certificate to the Keyring.            *
//* 5. Refresh RACF Classes required for digital certificates.        *
//* 6. Permit access to the Facility Class profiles.                  *
//*                                                                   *
//*                                                                   *
//* The following JCL demonstrates a sample implementation:           *
//* 1. Update all occurrences of "Userid-running-HSISANLO" to reflect *
//*    your TADz HTTPS environment.                                   *
//*                                                                   *
//* Do not change the RACF keyring 'TADZ_KEYRING' or label 'TADZCERT' *
//* unless you update the corresponding values in analyzer PARMLIB    *
//* member HSISANP2 and restart the Analyzer STC/Job.                 *
//*-------------------------------------------------------------------*
//RACFDEF    EXEC  PGM=IKJEFT01,DYNAMNBR=30
//SYSTSPRT   DD    SYSOUT=*
//SYSTSIN    DD    *
 PROF NOPREF

 RACDCERT DELETE(LABEL('TADZCERT'))
 RACDCERT ID(CMACN) DELRING(TADZ_KEYRING)

 SETROPTS CLASSACT(DIGTCERT,DIGTNMAP)
 SETROPTS RACLIST(DIGTCERT,DIGTNMAP)

 RACDCERT ID(Userid-running-HSISANLO) ADDRING(TADZ_KEYRING)

 RACDCERT ID(Userid-running-HSISANLO) GENCERT -
 SUBJECTSDN (CN('TADZCERT')                   -
 OU('Your Dept.')                             -
 C('US'))                                     -
 WITHLABEL('TADZCERT')

 RACDCERT ID(Userid-running-HSISANLO)         -
 CONNECT(ID(Userid-running-HSISANLO)          -
 LABEL('TADZCERT')                            -
 RING(TADZ_KEYRING)                           -
 DEFAULT                                      -
 USAGE(PERSONAL))

 RACDCERT ID(Userid-running-HSISANLO)         -
 CONNECT(ID(Userid-running-HSISANLO) CERTAUTH -
 LABEL('Entrust Secure Server Root CA')       -
 RING(TADZ_KEYRING)                           -
 USAGE(CERTAUTH))
```

```
 SETROPTS RACLIST(DIGTCERT,DIGTNMAP) REFRESH
/*
//PERMIT     EXEC  PGM=IKJEFT01,DYNAMNBR=30
//SYSTSPRT   DD    SYSOUT=*
//SYSTSIN    DD    *
 PROF NOPREF

  RDEL FACILITY IRR.DIGTCERT.LIST
  RDEL FACILITY IRR.DIGTCERT.LISTRING
  SETR RACLIST(FACILITY) REFRESH

  RDEFINE FACILITY IRR.DIGTCERT.LIST  UACC(NONE)
  RDEFINE FACILITY IRR.DIGTCERT.LISTRING  UACC(NONE)
  SETR RACLIST(FACILITY) REFRESH

  PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)     -
  ID(Userid-running-HSISANLO) AC(READ)

  PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) -
  ID(Userid-running-HSISANLO) AC(READ)

  SETR RACLIST(FACILITY) REFRESH
/*
```

Running the Analyzer in Batch mode

If you want to automate report generation, you can run the Analyzer in batch mode.

HSISANLB in JCLLIB contains sample JCL.

```
//*
//   SET OUTFMT=TXT
//*  SET OUTFMT=XLS
//*  SET OUTFMT=CSV
//*  SET OUTFMT=HTM
//*
//   SET OUTDSN=&SYSUID..TADZANLZ.&OUTFMT  Output dsn
//*
//ALLOC   EXEC PGM=IEFBR14
//OUTDSN   DD DISP=(MOD,CATLG),DSN=&OUTDSN,
//            DCB=(DSORG=PS,RECFM=VB,LRECL=1000,BLKSIZE=0),
//            UNIT=SYSALLDA,SPACE=(CYL,(5,10))
//*
//ANALYZER EXEC PGM=HSICANLZ
//STEPLIB  DD DISP=SHR,DSN=HSIDEV.V720.D111.SHSIMOD1
//         DD DISP=SHR,DSN=DB2V910.DE91.SDSNEXIT
//         DD DISP=SHR,DSN=DB2.V910.SDSNLOAD
//SYSPRINT DD SYSOUT=*,HOLD=YES,LRECL=500
//HSIANL1  DD DISP=SHR,DSN=HSIDEV.V720.D111.SHSIANL1
//HSIANL2  DD DISP=SHR,DSN=HSIDEV.V720.D111.SHSIANL2
//DSNAOINI DD DISP=SHR,DSN=MPRES.V750D111.PARMLIB(HSISCLI)
//HSICUST  DD DISP=SHR,DSN=MPRES.V750D111.PARMLIB(HSISANCQ)
//*HSINLS   DD DISP=SHR,DSN=HSIDEV.V720.D111.SHSIANL1(HSINLSJP)
//TPARAM   DD DUMMY
//OUTPUT1  DD DISP=OLD,DSN=&OUTDSN
//SYSIN    DD *
/asset/audit_trail
vendor      = IBM
showfeature = on
/*
```

The report name and parameters are specified in the SYSIN DD and the output goes to the OUTPUT1 DD.

The simplest way to know what report name and parameters to specify is to run the report first using Analyzer in online mode. At the bottom of every report, the report name and parameters are listed in the syntax needed for batch mode. You can cut and paste this syntax into the batch SYSIN DD.

Alternatively, you can directly type in the parameters. Wildcard filters have been enabled to assist in this case.

See "Analyzer report parameters" on page 221.

Controlling the Analyzer address space

The Analyzer supports the z/OS modify commands found in the table:

Table 24. z/OS modify commands

Command	Description
STOP	Stops the Analyzer address space. For example `/F HSISANLO,STOP` You can also issue this via the z/OS Stop command `/P HSISANLO`
REFRESH	Refresh Analyzer report templates and NLS text. For example `/F HSISANLO,REFRESH`. This is typically used to load new Custom queries
TRACE	Toggles on/off tracing. For example`/F HSISANLO,TRACE`. This should only be used when requested by IBM Support.

Analyzer custom queries

You can define custom queries in the Analyzer. This involves SQL queries (often based on supplied Analyzer queries) and the output is in standard Analyzer format.

HSISANCQ in PARMLIB has two sample custom queries:

```
</tadz_analyzer custom="Usage Data Summary"
                  desc="Summary of usage data gathered per system"

SELECT LP.FLPARNAME       AS "System"
     , MIN(UP.FFIRSTUSED) AS "First Date"
     , MAX(UP.FLASTUSED)  AS "Last Date"
     , SUM(UP.FEVENTCNT1) AS "Events"
  FROM TLPAR   AS LP
  JOIN TUSEPOV AS UP ON UP.FLPARID = LP.FLPARID
  WHERE LP.FLPARID >= 1
  GROUP BY LP.FLPARNAME
  ORDER BY LP.FLPARNAME


</tadz_analyzer>


<tadz_analyzer custom="User Appl Code Product Usage"
  desc="Product Usage Summary per Appl Code in 2-3 chars of userid">

SELECT SUBSTR(PUD.USERNAME, 2, 2)      AS "Appl Code"
     , P.VENDOR_NAME                   AS "Vendor"
     , P.PRODUCT_NAME                  AS "Product"
     , 'V'||CAST(P.VERSION AS CHAR(8)) AS "Version"
     , P.PID                           AS "PID"
     , COUNT(DISTINCT LP.FLPARID)      AS "Systems"
     , COUNT(DISTINCT PUD.USERNAME)    AS "UserIds"
```

```
     , COUNT(DISTINCT PUD.JOBNAME)     AS "Job Names"
     , MIN(PUD.START_DATE)             AS "First Used"
     , MAX(PUD.END_DATE)               AS "Last Used"
     , SUM(PUD.EVENT_CNT)              AS "Events"
FROM PRODUCT_USE_DETAIL AS PUD
  JOIN PRODUCT            AS P   ON P.SW_KEY   = PUD.VERSION_GUID
  JOIN TLPAR              AS LP  ON LP.FLPARID = PUD.FLPARID
  WHERE PUD.USERNAME IS NOT NULL
    AND PUD.USERNAME <> '  '
  GROUP BY SUBSTR(PUD.USERNAME, 2, 2)
         , P.VENDOR_NAME
         , P.PRODUCT_NAME
         , P.VERSION
         , P.PID
  ORDER BY SUBSTR(PUD.USERNAME, 2, 2)
         , UPPER(P.VENDOR_NAME)
         , UPPER(P.PRODUCT_NAME)
         , P.VERSION

</tadz_analyzer>
```

The simplest way to develop the SQL for a custom query is to:

- Run a standard Analyzer report that has similar data that you want to query.
- At the end of the query, you will see a line saying how many rows have been queried and a hyperlink on the DB2 Subsystem. Click the hyperlink and you will see the SQL used to generate the report. Select the SQL, right-click and select “copy”.
- Under Administration, run the SQL Select Query report. Right-click the SQL box and select “paste”. Click Submit to run the report.
- Modify the SQL as desired. Details about the table columns are documented in Appendix D of this manual.
- In TSO ISPF, edit HSISANCQ, and paste your custom SQL into a new report like the other two sample custom reports. You can remove the original samples if you want. Alternatively, create a new member and adjust the HSICUST DD accordingly.
- To see the new custom report you need to either restart Analyzer, or issue the Analyzer REFRESH modify command. For example `/F HSISANLO,REFRESH`.

If you define a HSINLS DD, it contains a Codepage setting. All text in the HSICUST DD is treated as being in the same Codepage as the HSINLS setting. This includes the actual SQL, so it is important that the Codepage is compatible with the Codepage your DB2 has been configured to use. This is particularly important for SQL statements that contain special characters such as concatenation bars (`||`), as these symbols are often sensitive to codepage differences.

(Optional) Installing Tivoli Common Reporting

Tivoli Common Reporting V2.1 is provided so that you can develop your own Cognos based custom reports. It is useful when combining external data sources with Tivoli Asset Discovery for z/OS data. Please note that there are no pre-configured reports in Tivoli Common Reporting V2.1. These reports are only provided in the Analyzer.

Tivoli Common Reporting is packaged with Tivoli Asset Discovery for z/OS, and requires a server running a Windows or Linux platform.

1. Install Tivoli Common Reporting V2.1.

For more information on how to install, see the relevant chapter in the *Tivoli Common Reporting User Guide*.

2. Install DB2 Connect™, Version 9. For more information about how to install, see *DB2 Connect Quick Beginnings for DB2 Connect Personal Edition, (GC23-5839-03)* Part 3, "Installing DB2 Connect Personal Edition". This product provides the ODBC drivers that are required for Tivoli Common Reporting connectivity to DB2 on the host.
3. Download the Tivoli Asset Discovery for z/OS Cognos based report package.

   Note: The supplied Cognos based report package contains only a Cognos Stage 1 model template. It does not contain any pre-configured reports.

   This section describes how to transfer the Tivoli Asset Discovery for z/OS Cognos based report package from the host system by using FTP. A Tivoli Asset Discovery for z/OS for z/OS administrator should have the necessary authorities to perform this task. To connect to the host system and copy the Tivoli Asset Discovery for z/OS for z/OS report package:

   a. Connect to the host system by using command-line FTP

      ```
      C:/> ftp hostname
      ```

   b. When prompted, enter your user name and password.
   c. Switch the FTP mode to binary mode.

      ```
      ftp > binary
      ```

   d. Download the report package member

      ```
      ftp > get 'hsi.SHSITCR1(HSICOG)' <path> TADz_model.zip
      ftp > bye
      ```

4. Copy the report package to the Tivoli Common Reporting server. The report package downloaded in the previous step must be placed in the folder: `C:\IBM\tivoli\tipv2Components\TCRComponent\cognos\deployment`. For information about importing the report package, see Appendix E, “Reporting with Tivoli Common Reporting,” on page 241.
5. For Tivoli Common Reporting to access the host data, two jobs are provided in Tivoli Asset Discovery for z/OS, which populate data on the host. These jobs are generated from the HSISCUST post installation customization job and are found in the JCLLIB.

   a. **HSISCOGT**. Submit this job to create and populate shared tables. To use the Cognos reporting engine, a set of cross product reporting tables must be defined and populated on the same DB2 subsystem as the one where the Tivoli Asset Discovery for z/OS Repository database is located. These tables are defined on a separate database and table space.

      This job will also populate the TIME_DIMENSION table based on the following parameters:

      SSID =
      DB2 subsystem name. Value assigned, as defined in job HSISCUST.

      START_ DATE =
      Defines the first day from which you would like to start populating data into the TIME_DIMENSION table. This field is populated at a daily level, meaning that if you have three records processed at different times on a day, all three records are condensed into a single record for that day.

DAYS =

Defines how many days you want to populate in the TIME_DIMENSION.

b. **HSISCOGA**. Submit this job to create aliases for Tivoli Asset Discovery for z/OS tables. This job creates aliases for the 8 Asset tables that are referenced by the Cognos Stage 1 model template.

Chapter 6. Utilities

This section describes the different functional tasks that need to be performed during the product life cycle. They are:

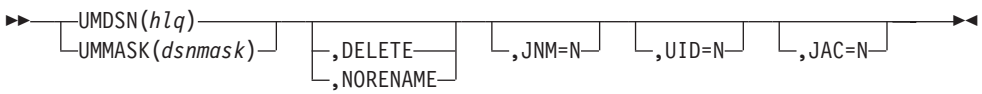
ZCAT utility

The ZCAT utility is used to concatenate and condense Usage Monitor data sets. The resulting file is then processed by the Usage Import program. The condensation of data produced by the Usage Monitor results in improved Usage Import performance and saving of storage space. The Usage Monitor started task produces at least one usage data set per day. Work flow can be designed to run ZCAT on the data sets on a weekly, fortnightly, or monthly basis before processing by the Usage Import. A weekly run is recommended, but depends on the amount of data that is produced and processed at your site. The Usage Monitor collects a large amount of detail about which job, account ID, and user ID are using a particular module of a given library on a given date. This information is output into multiple files that are produced on a daily basis. The ZCAT utility condenses the files as follows:

- Usage data across multiple files is condensed to a monthly granularity, as are the records stored in the Repository database.
- Redundant records across files, and records that are not stored in the database, are not included.
- Optionally, condensation can occur across various user IDs, job names, or account ID details. For further details see the description of condensation options listed in "Optional condensation parameters" on page 90.
- The ZCAT output file is compressed and ready to be transmitted for Usage Import processing.

Running the ZCAT utility

ZCAT syntax



ZCAT parameters

Mandatory Parameters

The UMDSN or the UMMASK parameter must be specified.

UMDSN(*hlq*)
hlq is the Usage Monitor data set high-level qualifier. When the **UMDSN** parameter is specified, ZCAT concatenates all data sets having names of hlq.Dyyyyddd.Thhmmsst where yyyyddd and hhmmsst are the timestamp patterns of data sets produced by the Usage Monitor. The *hlq* can contain wildcard characters of percent or asterisk. The percent character denotes a single character mask, and the asterisk character denotes all characters. For example UMDSN(TADZ.**) would search for all data set names of TADZ.**.D%%%%%%%.T%%%%%%%

UMMASK(*dsnmask*)
dsnmask is the full dsn mask search criteria. It can be used to search for a pattern of files that differ from the files produced by the Usage Monitor. This parameter is useful if the files produced by the Usage Monitor have been renamed, but still need processing. Specifying UMMASK(hlq.D%%%%%%%.T %%%%%%%) is equivalent to specifying UMDSN(hlq)

Optional parameters

One or more optional parameters can follow the mandatory parameters.

DELETE
Delete the input data sets after the output data set is successfully generated. The default is to retain the input data sets, which are renamed by default.

NORENAME
Do not rename input data sets from hlq.D*.T* to hlq.D*.S* after the output data set is successfully generated. The default is to rename these input data sets to stop them being reprocessed by the ZCAT utility. Use this option only to rename the data sets before further ZCAT processing. This option stops double counting of usage data.

The RENAME option is ignored, if **DELETE** is also specified.

Optional condensation parameters

Improvements in performance and data storage space are gained by using the ZCAT utility options to carry out further condensation of data, ignoring data differences that are not important at your site, and do not appear in your regular reporting. You can still point the Usage Monitor File Detail Report to the saved archive of the concatenated detail file (ZCATDETL), or to the Usage Monitor output files. ZCATDETL is produced by the ZCAT utility.

JNM=*N*
Condense data for different job names to one of the following generic names based on the job type: -STC-, -JOB-, -UID- or -SYS-. The default is to retain the job name, and to condense data that belongs to the same job name only.

UID=*N*
Condense data for different user IDs, which are converted to blank. The default is to retain the user ID, and to condense data that belongs to the same user ID only.

JAC=*N*

Condense data for different job account codes which are converted to blank. The default is to retain the job account code, and to condense data that belongs to the same job account code only

Note: Due to the various consolidating options, records that are ignored in the ZCATOUT data set are still written to the ZCATDETL output data set, which can be retained for archiving.

DD statements

ZCATOUT

Specifies the name of the ZCAT output data set. This data set can then be used as the input to the Usage Import program, where usage details are imported into the database. If the ZCATOUT DD card is omitted, ZCAT by default writes to a data set having the name hlq.Dyyyyddd.Uhhmmsst (U instead of T implied by the high level qualifier (hlq) option for input data sets), where yyyyddd and hhmmsst refer to the date and time timestamp of the first processed input data set.

ZCATDETL

If the ZCATDETL DD is allocated, all records are written to this data set. This data set includes any non condensed and non diagnostic data that is not written to the ZCATOUT data set. It enables the Job name, user ID, and job account details (which are ignored due to ZCAT options and are not written to the ZCATOUT file) to be archived into the detail file.

The ZCATDETL and ZCATOUT data sets are compressed by the ZCAT utility.

```
//ZCAT     EXEC PGM=HSICZCAT,PARM='UMDSN(TADZ.**),JNM=N'
//STEPLIB  DD  DISP=SHR,DSN=TADZ.V750.SHSIMOD1
//ZCATOUT  DD  DSN=&SYSUID..TADZ.ZCATOUT,
//             DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//             DCB=(DSORG=PS,RECFM=VB,LRECL=27994,BLKSIZE=27998)
//ZCATDETL DD  DSN=&SYSUID..TADZ.ZCATDETL,
//             DISP=(NEW,CATLG),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//             DCB=(DSORG=PS,RECFM=VB,LRECL=27994,BLKSIZE=27998)
//DEBUG    DD  SYSOUT=*,HOLD=YES,LRECL=200 //SYSPRINT DD  SYSOUT=*,HOLD=YES
//SYSERR   DD  SYSOUT=*,HOLD=YES
```

In this example, all data sets having names of TADZ.*.*.D%%%%%%%.T %%%%%%% are processed due to the **UMDSN** parameter. The condensed output is written to sysuid.TADZ.ZCATOUT where the SYSUID system symbol is the user ID of the person submitting the job. This file is then transmitted for Usage Import processing. The **JNM=N** parameter instructs the utility to condense job names and ignore the original job name distinction. All valid records are written to the ZCATDETL DD card sysuid.TADZ.CATDETL, which is then archived for reference purposes.

Product Tagging utility

The Product Tagging utility, also called the Tagger, provides a means of identifying software which has not been predefined in the Global Knowledge Base (GKB). It is also used to supersede GKB entries, without changing the GKB contents.

Product Tagging is a manual process which requires you to provide details of the product name, vendor, and location of the programs. The Tagger scans the programs, by using the same method as the Inquisitor, and records the results in dedicated program members.

The control statements that describe which licensed programs are to be tagged are found in the SYSIN file. This file contains details of the program name, vendor name, product identifier, and version. The program library which contains the software to be tagged is allocated to the SYSLIB file.

You can have only one set of identifying attributes for each program name. If conflicting attributes are found for one or more program names, the Tagger issues a message and stops.

Information about all discovered programs relating to the nominated product is compiled into a single object module. This module is written to the scanned library allocated to SYSLIB or to the program library allocated to the optional HSIREDIR file. Using the HSIREDIR file, you can nominate to keep all tag data separate from licensed program software. The HSIREDIR file data sets must be included in the normal Inquisitor scan processing, even if these data sets contain no other program.

The tag data members created by the Tagger are recognized by the Inquisitor (by their SSI value) during normal program library scanning. The Inquisitor extracts the tag data from the member contents and writes it to an output file. The Inquisitor Import process uses these program tags to maintain entries for the programs in the Local Knowledge Base. The Match Engine can then accurately identify the tagged product level, regardless of which library the product is deployed to and which system the data is collected from.

The Tagger scans a single library and tags a single software product, or optional feature of a product, in each run. For products with multiple program libraries, each library is processed in a separate Tagger job or step. To ensure effective software identification by the Match Engine as it processes each library, use the OPTION statement to differentiate the identification entities between the different libraries of a product. Distribution libraries should not be tagged.

You can override the default output member name of @HSIPTAG by specifying a TAGMEM statement. All Tagger output members are flagged with an SSI value of X'D7E3C1C7', which is 'PTAG' in EBCDIC.

If there is no preexisting member of the same name, a new program member is created by the Tagger to contain the tag data. If a member exists, the new tag data is added to the preexisting data pertaining to other products or optional features. Any data pertaining to the same software identified by {VENDOR + PRODUCT + OPTION + VERSION} is replaced. The data relating to each software piece resides in its own control section. Tag data members contain no executable code, and are bound with the only loadable attribute. These data members are bound as reentrant, with a residence mode of ANY, to minimize the impact of being placed in a library which is loaded into the Link Pack Area.

To erase the effects of Tagger processing, delete the tag data members, which are identified by their SSI value. If you are using ISPF, employ the **SORT SSI** member list command.

The software processed in a Tagger run has a key of {VENDOR + PRODUCT + OPTION + VERSION}. If non-key data items, such as the values specified in the PPNUM or LICENSED statements are incorrect, you can correct them by fixing the input statement values and rerunning the Tagger. This action replaces all non-key tag data. However, if a key data item is incorrect, it will not be erased by running the Tagger with the correct data.

If you are processing libraries that are not dedicated to a single licensed program, use member name masking to prevent tagging programs not related to that product. Some installations place multiple software products in a combined common library. If the products are tagged before they are combined, different tag data member names must be used.

Running the Tagger

To run the Tagger, use the job HSISPTAG in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

Tagger control statements

Control statements can be input by using the SYSIN file. General syntax rules are:

- Fixed length, variable length, and undefined record formats are processed.
- Short records are extended to 72 bytes of data, with blanks if necessary.
- Only the first 72 bytes of data for each record are processed by the Tagger.
- Records beginning with an asterisk are treated as comments and do not alter continuation status.
- The first nonblanks of a statement must identify the statement type.
- One or more blanks must follow the statement type.
- A statement with no value or operand specified is invalid.
- For statement types other than SELECT, the specified value is deemed to start with the first nonblank after the statement type name.
- Statements can be placed in any order. All statements are processed before any tagging activity commences.
- SELECT is the only statement type which can be supplied more than once in an input file.
- SELECT is the only statement type which can be continued over more than one record.

Details of all Tagger statement types are found in the following table:

Table 25. Tagger statement types

Statement Type	Value	Default Value	Required	Maximum length
VENDOR	Vendor name	-	Yes	30 bytes
PRODUCT	Product name	-	Yes	50 bytes
PPNUM	Licensed program number	blanks	No	16 bytes
OPTION	Optional feature name	BASE	No	30 bytes
VERSION	Software level	-	Yes	8 bytes
LICENSED	Separately licensed feature? (YES or NO)	NO	No	3 bytes
TAGMEM	Output member name	@HSIPTAG	No	8 bytes
SELECT	Program name filter	PGM(*)	No	8 bytes per mask

SELECT is not a value-oriented statement type. It has operands which have values specified in parentheses. The PROGRAM or PGM inclusion operand can be abbreviated to P. The XPROGRAM or XPGM exclusion operand can be abbreviated to XP.

The Tagger stops parsing a SELECT record and the current statement continues on to the next record whenever a continuation character is encountered. Valid continuation characters are plus and hyphen. A continuation cannot occur within an operand name, or a value mask.

SELECT syntax

member-mask
A string up to 8 bytes in length, representing one or more possible member names of a PDS or PDSE. Use a percent sign to indicate that any single character is to be considered a match in the exact location of the compared character string. Use an asterisk to indicate that any zero or more characters are a match.

Example 1

A company called ISV has created a build of several programs (build 97) it is developing under the Swisho4U brand. The data sets created by this build have their own disk volume called BLD097. The tag data is to be redirected to a data set dedicated for this purpose.

```
//STEP1   EXEC PGM=HSITAGP
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=S4U.LOADLIB,DISP=SHR,UNIT=3390,VOL=SER=BLD097
//HSIREDIR DD  DSN=S4U.TAGLIB,DISP=SHR
//SYSIN    DD  *
VENDOR   ISV
PRODUCT  Swisho4U
VERSION  BUILD097
/*
```

Example 2

The BigBiz Inc. data center is about to deploy the contractor data processing component for Version 4 Release 2 of its internally developed human resources application called HU-MAN. The software is tagged in its own library, but the default tag member name is not used in case it is later loaded into a program library common to several applications. All programs in HU-MAN have names beginning with HU, but the contractor component is the only component which has program names beginning with HUC. The relevant program library can be accessed by using the catalog.

```
//TAGRUN  EXEC PGM=HSITAGP
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=HUMAN.V4R2M0.LOAD,DISP=SHR
//SYSIN    DD  *
VENDOR   BIGBIZ INCORPORATED
PRODUCT  HU-MAN Human Resources Management
OPTION   Contractor Handling
```

```
VERSION  04.02.00
TAGMEM   HUMANT@G
SELECT   PGM(HUC*)
/*
```

Example 3

Version 1.5 of the product MVSBLOAT from MiscWare has been deployed on a system which has a dedicated tag data library called SYS2.TAGLIB. Link list programs for the product have been placed in SYS2.LINKLIB and ISPF application modules have been placed in SYS2.ISPLLIB. The product does not have optional features, but only the base component installed. All the installed programs have names beginning with MVSB. The OPTION statement is used to ensure that the contents of each library can be identified by the Match Engine.

```
//STEP1   EXEC PGM=HSITAGP
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS2.LINKLIB,DISP=SHR
//HSIREDIR DD  DSN=SYS2.TAGLIB,DISP=SHR
//SYSIN    DD  *
VENDOR  MiscWare
PRODUCT MVSBLOAT
OPTION  BASE (Batch)
VERSION 01.05.00
TAGMEM  $$OEMTAG
SELECT  PGM(MVSB*)
/*
//STEP2   EXEC PGM=HSITAGP
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS2.ISPLLIB,DISP=SHR
//HSIREDIR DD  DSN=SYS2.TAGLIB,DISP=SHR
//SYSIN    DD  *
VENDOR  MiscWare
PRODUCT MVSBLOAT
OPTION  BASE (Dialogs)
VERSION 01.05.00
TAGMEM  $$OEMTAG
SELECT  PGM(MVSB*)
/*
```

Usage Summary

Usage Summary is the process of summarizing usage data in the Repository. This process deletes detailed usage records and creates monthly summary records, by reducing the number of DB2 rows used to represent your old data. After each Usage Summary, the usage data is aggregated to update the Asset tables.

To minimize space utilization and improve SQL query performance, it is recommended that you keep detailed module usage data for the last three months and summarize all detailed module usage data older than three months. It is also recommended to delete summarized module usage data older than 18 months. Please see job HSISUDEL (Usage Deletion) for more details.

If you have not run the Usage Summary job for some time, then select a period of a few months at a time, in order to keep the run times down to a reasonable time.

Running Usage Summary

To run the Usage Summary, use the job HSISUSUM, in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

TPARAM parameters

COMMIT=
Default is 1000. Number of records stored before issuing of COMMIT.

DSN= DB2 location. Value assigned, as defined in job HSISCUST.

SUMBY=
Use this parameter to condense the usage data.

SUMBY=1
Data is summarized at the Product, LPAR, Period, Module ID, and Job Type level.

User ID and job ID distinctions are ignored. Instead of Job IDs, events are attributed to Job Types (BATCH, TSO, DB2...).

SUMBY=2
Data is summarized at Product, LPAR, Period, Job ID, User ID level.

Load module and program names are ignored.

SUMBY=3
The rules for SUMBY=1 and SUMBY=2 apply.

Data is summarized by Product, LPAR, Period, Job Type.

KEEPDETAIL=
Default is 2. Number of months prior to the current month for which usage records are not summarized. Prior usage records are summarized. If KEEPDETAIL=0 is specified, all usage records, excluding those records for the current month, are summarized.

FIRSTDATE=
Start of the first date range. This is in the form YYYYMM. Only complete months are chosen.

LASTDATE=
End of the last date range. This is in the form YYYYMM.

Note: The date range of summarization is inclusive of the month specified in the FIRSTDATE and LASTDATE parameters.

MINUSAGETHRESHOLD
Default is 1000. Sets a value for Usage Summary to ignore summarization of usage records. If this parameter is set to 1000, then any product with a usage count of 1000 or less for any given month, does not have its usage records summarized. This allows you to view the usage records for low usage products.

REPSCHEMA=
Repository qualifier. Name of qualifier is *&DB*.

Usage Deletion

Usage Deletion is the process of deleting usage data in the Repository. This process deletes detailed, summarized and aggregated usage data for the period specified for all systems in the Repository. After each Usage Deletion, usage data is aggregated to update the Asset tables.

To minimize space utilization and improve SQL query performance, it is recommended that you keep no more than 3 months of detailed module usage

data and 13 months of aggregated product usage data. If you want to keep more than 3 months of detailed module usage data, then run job HSISUSUM (Usage Summary) to summarize the detailed module usage data older than three months. See job HSISUSUM for more details.

If you have not run the Usage Deletion job for some time, then select a period of a few months, in order to keep the run times down to a reasonable time.

Running Usage Deletion

To run the Usage Deletion, use the job HSISUDEL, in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

TPARAM parameters

COMMIT=
Default is 1000. Number of records stored before issuing of COMMIT.

DSN= DB2 location. Value assigned, as defined in job HSISCUST.

KEEPDETAIL=
Default is 2. Number of months prior to the current month for which detailed and summarized module usage data are kept. KEEPDETAIL=0 means all detailed and summarized module usage data excluding those from the current month are deleted.

KEEPAGGR=
Default is 12. Number of months prior to the current month for which aggregated product usage data are kept. KEEPAGGR=0 means all aggregated product usage data, excluding those from the current month are deleted.

FIRSTDATE=
Start of the first date range. This is in the form YYYYMM. Only complete months are chosen.

LASTDATE=
End of the last date range. This is in the form YYYYMM.

Note: The date range of deletion is inclusive of the month specified in the FIRSTDATE and LASTDATE parameters.

REPSCHEMA=
Repository qualifier. Name of qualifier is *&DB*.

SID= System Identifier of system or LPAR for which usage should be deleted.

Note: If KEEPDETAIL is set to a value, then FIRSTDATE / LASTDATE will be ignored. If detailed usage data are to be deleted within a certain date range, then comment out KEEPDETAIL and define dates for FIRSTDATE / LASTDATE. For further details, please see comments described in job HSISUDEL.

High-level qualifier listing for Usage Monitor

This job creates a high-level qualifier listing for Usage Monitor.

Because large amounts of usage data are collected, it is best practice to monitor libraries that are identified in the high-level qualifier listing.

This utility scans the Repository to create a list of high-level qualifiers for products that are to be identified. Listed here are some examples that exclude all usage, but include some usage for the specified high-level qualifiers:

```
XDS(*)
IDS(DB2.*)
IDS(IMS.*)
IDS(CICS.*)
IDS(SYS1.*)
```

This process is automated in the Inquisitor Import job. The high-level qualifier listing is written to a data set, and this data set is concatenated to the Usage Monitor control file, HSIZIN.

Running high-level qualifier for Usage Monitor

To run the High Level Qualifier for the Usage Monitor utility, use the job HSISLLST in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

TPARAM parameters

DSN= DB2 location. Value assigned, as defined in HSISCUST.

REPSCHEMA=
Repository qualifier. Name of qualifier is *&DB*.

COMMIT=
Default is 1000. Number of records stored before issuing a COMMIT.

TPARAM table update

In certain situations, the TPARAM table in the Repository might have been set to an inconsistent state, due to failures in jobs that update the Repository tables. To rectify this inconsistent state, a parameter in the TPARAM table has to be reset.

Running TPARAM table update

To run the TPARAM table update, use the job HSISTPRM, in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

SYSIN parameter

```
UPDATE &DB.TPARAM SET FVALUE = '0' WHERE FKEY = 'PROCRUN';
```

SCRT import utility

The Tivoli Asset Discovery for z/OS SCRT import utility reads data created by the IBM Subcapacity Reporting Tool. This comes in the form of CSV formatted output files, and is used to populate several DB2 tables in the Repository. This information can then be queried by Tivoli Common Reporting to provide trending of SCRT data, as well as comparing this with the corresponding usage data.

Running SCRT import utility

To run the SCRT import utility, use the job HSISSCRT, in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

INPUT

DDNAME CSVIN, which contains the CSV output from the IBM SCRT tool. This may be from a data set with DSORG of PS or PO. Binary uploaded CSV files are supported.

DDNAME SIDMAP is used to map duplicate SMFIDs to a unique SID. The SCRT Import Utility is used to handle data where the same SMFID is used on multiple machines concurrently.

Map SMFID on specific machines to your desired SID. As described in this example, when processing data for CPU serial 11111, SMFID IP01, use SID QIP1, and so on.

```
//SIDMAP   DD *
11111-IP01=QIP1
11111-IP02=QIP2
11111-IP03=QIP3
/*
```

CPU serial
5 alphanumeric characters

SMFID
1 to 4 alphanumeric characters

Unique SID
1 to 4 alphanumeric characters. This must be the same as the SID value being used by the Usage Monitor for that z/OS system.

OUTPUT

Several DB2 tables are populated from the data contained in CSVIN, including NODE, NODE_CAPACITY, and PRODUCT_NODE_CAPACITY. Ensure that the CSVIN DD points to the .CSV output file created by the IBM SCRT tool. This may be a DSORG=PO or PS data set.

TPARAM parameters

SSID=
DB2 subsystem name. Value assigned, as defined in job HSISCUST

REPSCHEMA=
Repository qualifier. Name of qualifier is *&DB*.

GKB=
Global Knowledge Base qualifier. Name of qualifier is *&DBGKB*_GKB7

SMF Scanner

The SMF Scanner utility provides you with the ability to get historical usage information from already existing SMF data. Using this SMF data means you can view trending results from before Tivoli Asset Discovery for z/OS is installed.

To start the process you need to run two jobs to capture scanned data (Inquisitor) and historical usage data (SMF Scanner). The output from the SMF Scanner (usage data) can then be processed to produce historical trending.

A sample job HSISIBM can take a file from either the Inquisitor or the Usage Monitor and filter out non-IBM programs. You might use this function when sending data to IBM Support for diagnosis.

The output of the SMF Scanner may also be used as input to HSISIBM.

Note: The SMF Scanner only tracks usage of the Job Step EXEC PGM modules and does not include modules that have been invoked from within the task.

Running SMF scanner utility

To run the SMF scanner utility, use the job HSISSMF in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

XML export utility

This utility extracts information in XML format for customers to import into the Tivoli Asset Management for IT product. The extracted information can be either:

- Catalog of products installed in your system.
- Catalog of products defined in Global Knowledge Base.

Running XML export utility

To run the XML export utility, use the job HSISKBT, in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

The output XML file generated from this utility needs to be transferred by FTP to a distributed environment and then loaded into Tivoli Asset Management for IT (TAMIT). Also the XML file needs to be translated from EBCDIC to ASCII. Following is an example on how to perform this transfer.

1. C:\temp ftp *host name* (Using command line FTP connect to the host system)
2. When prompted enter your user name and password.
3. ftp > quote type a (set to ASCII)
4. ftp > get 'hsiinst.SWKBT.XML' C:\XML.FILE
   (ftp file from host to local workstation)
5. ftp > exit
   If you need to ftp non ASCII characters, then issue "ENCODING" statement as this extra command before the GET command :
   quote site ENCODING=MBCS MBDATACONN=(IBM-939,UTF-8)
   (this example is for a Japanese codepage)

TPARAM parameters

SSID=
DB2 subsystem name. Value assigned, as defined in job HSISCUST.

SCHEMA=
Repository qualifier or Global Knowledge Base qualifier.

a) Using the Repository qualifier value means that a catalog of products installed on your site is selected

b) You can also use the Global Knowledge Base qualifier value. This would mean that a catalog of all products defined in the Global Knowledge Base is selected.

Chapter 7. Globalization

Customizing tasks for Japanese messages

MVS™ Message Service (MMS) message information is provided for the Japanese language.

MMS message information for each language is contained in the following PDS:

- hsi.SHSIMJPN (Japanese)

The MMS message file must be compiled using the MMS message compiler. See the *MVS Assembler Services Guide* for more information on loading MMS messages.

To compile MMS files, use the job HSISMCMP in the JCLLIB. This job is generated from the HSISCUST post-installation customization job.

You must install the latest system runtime message file.

Here is an example of how to create a new Japanese system runtime message file and install it:

1. Compile MMS files into system runtime message file hsi.MMSJPN99
2. Create a new entry in PARMLIB called MMSLSTJ9

   ```
   DEFAULTS LANGCODE(JPN)
   LANGUAGE LANGCODE(JPN) DSN(SYS2.MMSJPN99) CONFIG(CNLJPN00)
   ```

3. Install the latest system runtime message file using this MVS system command

   ```
   SET MMS=J9
   ```

Note:

The components of the product listed here do not use the MVS message service MMS:

- Inquisitor Import
- Usage Import
- Usage Summary
- Usage Deletion

Japanese customers can enable Japanese messages to the MSGFILE for these components if the Language Environment NATLANG(JPN) option is in effect. For more information on the Language Environment MSGFILE and NATLANG options, refer to the *Language Environment Programming Reference (SA22–7562).* For more information about specifying Language Environment runtime options, refer to the *Language Environment Programming Guide (SA22–7561).* For more information about setting NATLANG(JPN) as an installation default, refer to the *Language Environment Customization (SA22-7564).* When the NATLANG(JPN) option is not in effect, the components listed here always output messages in mixed case English.

Enabling Analyzer for Japanese

To enable the Analyzer for Japanese, update the DD statements in the Analyzer started task HSIJANLO as follows:

```
//HSICUST  hsi.SHSIPARM(HSISANCJ)
//HSINLS   hsi.SHSIANL1(HSINLSJP)
```

If you wish to customize your own reports, copy `hsi.SHSIPARM(HSISANCJ)` to `hsiinst.PARMLIB(HSISANCJ)` and make changes directly in `hsiinst.PARMLIB(HSISANCJ)`

Update the Analyzer started task HSIJANLO as follows:

```
//HSICUST hsiinst.PARMLIB(HSISANCJ)
```

Appendix A. System messages

This section lists the various messages output by Tivoli Asset Discovery for z/OS.

HSIA - Automation Server messages

Return codes

0	No errors encountered. All requests processed successfully.
16	Unrecoverable error. No requests processed. SYSIN or HSIPZIP or INQSOUT File cannot be used, or unsupported Operating System.

Message suffix codes

Suffix	Meaning	Raises Minimum Cond Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16,20

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIA001E EXPECTED CLOSE PARENTHESIS WAS NOT FOUND IN INPUT RECORD

Explanation: Parsing did not detect the expected right parenthesis.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA002E EXPECTED VALUE FOR *parm* NOT FOUND BEFORE THE CLOSE PARENTHESIS

Explanation: No subparameter value was specified within the parentheses.

In the message text:

parm
name of parameter being processed.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA003E THE *parm* PARAMETER IS NOT RECOGNIZED

Explanation: A parameter was detected which is not valid for the type of statement being processed.

In the message text:

parm
name of parameter being processed.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO

HSIA004E THE VALUE SPECIFIED FOR *parm* IS INVALID

Explanation: The named parameter had a value which did not conform to syntax requirements.

In the message text:

parm
name of parameter being processed.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO

HSIA005E NO "FTP" OR "JOB" STATEMENT BEFORE "DSN" STATEMENT

Explanation: There is no preceding action to relate the dsname mask to.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA006E *parm* IS AN UNRECOGNIZED STATEMENT TYPE

Explanation: A statement type other than FTP, JOB, or DSN was specified.

In the message text:

parm
name of parameter.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA007E TERMINATING - AUTOMATION SERVER IS ALREADY ACTIVE

Explanation: The Automation Server is already running. Only one concurrent copy can run in an operating system image.

System action: Program terminates with condition code 16. The established Automation Server continues.

Operator response: None.

System programmer response: None.

Module: HSIAUTO

HSIA008E TERMINATING - COULD NOT INITIALISE WITH BAD PARAMETERS

Explanation: The HSIAPARM contents contained an error so the Automation Server could not initialize.

System action: The program terminates with condition code 16.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA009E REFRESH IGNORED - COULD NOT PROCESS BAD PARAMETERS

Explanation: The HSIAPARM contents contained an error so the Automation Server could not update its operational parameters.

System action: Terminates the processing of HSIAPARM contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA010E NO FUNCTIONS WERE REQUESTED

Explanation: No actions were specified. The Automation Server has no work to do.

System action: Terminates the processing of HSIAPARM contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA011E NO DATA SET NAME MASKS WERE SPECIFIED

Explanation: No work was requested. The Automation Server has no work to do.

System action: Terminates the processing of HSIAPARM contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA012E NUMBER OF ACTIONS EXCEEDS MAXIMUM OF 1000

Explanation: Too many actions were requested.

System action: Terminates the processing of HSIAPARM contents.

Operator response: Reduce the number of actions specified.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA013E NUMBER OF DATA SET NAME MASKS EXCEEDS THE MAXIMUM OF 2000

Explanation: Too many dataset name masks were specified.

System action: Terminates the processing of HSIAPARM contents.

Operator response: Reduce the number of dataset name masks.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA014E MEMBER *mbr* WAS NOT FOUND IN THE HSIACNTL FILE

Explanation: Member HSIAPARM was found to be missing from the HSIACNTL file.

In the message text:

mbr
name of missing member.

System action: Terminates the processing of the member. If the member is HSIAPARM the Automation Server terminates. For template members the Automation Server continues processing.

Operator response: Create the required member in the HSIACNTL library.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA015E INPUT LOGICAL RECORD LENGTH WAS NOT 80

Explanation: A record was read from the HSIACNTL library which was not 80 bytes long.

System action: The program terminates and takes no actions.

Operator response: Ensure the HSIACNTL library only contains fixed length 80 byte records.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA016E EXPECTED OPEN PARENTHESIS WAS NOT FOUND IN INPUT RECORD

Explanation: Parsing did not detect the expected left parenthesis.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO

HSIA017E VALUE SPECIFIED FOR *parm* IS TOO LONG

Explanation: A parameter value was specified which has a length greater than the maximum allowed for the named parameter.

In the message text:

parm
name of parameter.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO HSIADSN

HSIA018E THE "NOTA" VALUE IS LESS THAN THE "NOTB" VALUE

Explanation: The action can never be performed because all days of the month have been excluded by the combination of the NOTA (not after) and NOTB (not before) specifications.

System action: Terminates the processing of the HSIAPARM member contents.

Operator response: Correct the HSIAPARM member contents.

System programmer response: None.

Module: HSIAUTO

HSIA999U HSIMSG/HSIAMSG FAILURE - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Ensure Joblib/Steplib contains the library where the HSIAMSG message module resides. If you cannot resolve this issue then contact IBM support.

Module: HSIAUTO :emsgl.

HSIF - Conversion messages

Return codes

0	No errors encountered. All requests processed successfully.
4	I/O error in one or more program libraries.
8	Error - Incomplete data. Processing continues. OPEN or other system service error.
12	Syntax error.
16	Unrecoverable error.
20	Disastrous error. No requests processed. SYSPRINT file cannot be used.

Message suffix codes

Suffix	Meaning	Raises Minimum Condition Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIF000U PROCESSING TERMINATED - NO USABLE SYSPRINT FILE

Explanation: The OPEN of the SYSPRINT file failed. Note: This message is issued by WTO with ROUTCDE=(2,11).

System action: Terminates with a condition code of 20.

Operator response: Ensure a usable SYSPRINT file is allocated.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF001S PROCESSING TERMINATED - STORAGE OBTAIN ENCOUNTERED AN ERROR, RC=*rc*

Explanation: Unable to acquire required amount of storage.

In the message text:

rc return code from STORAGE macro.

System action: Terminates with a condition code of 16.

Operator response: None.

System programmer response: Try increasing the region size specified in the region parameter on the JOB or EXEC statement in the JCL for the job. For additional information, examine the return code shown in the message and use for problem determination/resolution.

Module: HSIIM2D

HSIF002S PROCESSING TERMINATED - DD FOR SYSPRINT MISSING

Explanation: DD statement missing for SYSPRINT. Note: This message is issued by WTO with ROUTCDE=(2,11).

System action: Terminates with a condition code of 20.

Operator response: Ensure DD statement in the JCL is correct.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF004S PROCESSING TERMINATED - CANNOT OPEN FILE "*file*"

Explanation: The OPEN of the file failed.

In the message text:

file
name of file that failed the OPEN request.

System action: Terminates with a condition code of 16.

Operator response: Ensure a usable file is allocated.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF005S PROCESSING TERMINATED - DD FOR FILENAME "*file*" MISSING

Explanation: DD statement missing for the file.

In the message text:

file
name of file with missing DD statement.

System action: Terminates with a condition code of 16.

Operator response: Ensure DD statement in the JCL is correct.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF007S PROCESSING TERMINATED - INVALID SYSIN PARAMETER

Explanation: Parsing detected an invalid parameter.

System action: Terminates with a condition code of 12.

Operator response: The final parameter displayed in the SYSPRINT report is in error. Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF008S PROCESSING TERMINATED - INVALID SYSIN PLX= PARAMETER. MUST BE Y OR N

Explanation: The SYSIN PLX= parameter must be set to Y or N.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF009S PROCESSING TERMINATED - INVALID SYSIN SYSPLEX= PARAMETER. THE SYSPLEX NAME MUST BE ENTERED

Explanation: The SYSIN SYSPLEX NAME must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF010S PROCESSING TERMINATED - INVALID SYSIN SSID= PARAMETER. THE SMF SYSTEM IDENTIFIER MUST BE ENTERED

Explanation: The SYSIN SSID= parameter must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF011S PROCESSING TERMINATED - INVALID SYSIN SSMF= PARAMETER. THE SMF SYSTEM IDENTIFIER MUST BE ENTERED

Explanation: The SYSIN SSMF= parameter must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D HSIIS2D

HSIF012S PROCESSING TERMINATED - INVALID SYSIN LPARNAME= PARAMETER. THE LPAR NAME MUST BE ENTERED

Explanation: The SYSIN LPAR NAME must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF013S PROCESSING TERMINATED - INVALID SYSIN SERIAL= PARAMETER. THE SERIAL NUMBER MUST BE ENTERED

Explanation: The SYSIN SERIAL NUMBER parameter must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF014S PROCESSING TERMINATED - INVALID SYSIN HWTYPE= PARAMETER. THE HARDWARE TYPE MUST BE ENTERED

Explanation: The SYSIN HARDWARE TYPE parameter must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF015S PROCESSING TERMINATED - INVALID SYSIN HWMODEL= PARAMETER. THE HARDWARE MODEL MUST BE ENTERED

Explanation: The SYSIN HARDWARE MODEL must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF016S PROCESSING TERMINATED - INVALID SYSIN UNM= PARAMETER. MUST BE Y OR N

Explanation: The SYSIN UNM= parameter must be set to Y or N.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF017S PROCESSING TERMINATED - INVALID SYSIN JAC= PARAMETER. MUST BE Y OR N

Explanation: The SYSIN JAC= parameter must be set to Y or N.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF018S PROCESSING TERMINATED - INVALID SYSIN PLANT= PARAMETER. THE PLANT MUST BE ENTERED

Explanation: The SYSIN PLANT must be entered.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF019S PROCESSING TERMINATED - OUTPUT HEADER NOT CREATED. NO STATISTICS OR CAPACITY RECORD IN INPUT FILE

Explanation: The input file contains no Statistics or Capacity record.

System action: Terminates with a condition code of 16.

Operator response: Ensure a usable input file exists and rerun the program.

System programmer response: None.

Module: HSIIM2D

HSIF020S PROCESSING TERMINATED - HSIMONZP *func* FAILURE, RC=*rc*

Explanation: HSIMONZP file processing failure.

In the message text:

func
file processing in error.

rc return code from HSIMONZP.

System action: Terminates with a condition code of 16.

Operator response: Ensure the correct DD statements exist for the file and rerun the program. For additional information, examine the return code shown in the message and use for problem determination/resolution.

System programmer response: None.

Module: HSIIM2D

HSIF021S PROCESSING TERMINATED - NO DD STATEMENT DETECTED FOR HSIINQOT OR HSIINQZP

Explanation: At least one of the DD statements must be included.

System action: Terminates with a condition code of 16.

Operator response: Correct the JCL and rerun the program.

System programmer response: None.

Module: HSIIS2D

HSIF022S PROCESSING TERMINATED - *func* FAILURE, RC=*rc*

Explanation: Processing failure.

In the message text:

func
processing in error.

rc return code.

System action: Terminates with a condition code of 16.

Operator response: None.

System programmer response: Contact IBM support.

Module: HSIIS2D

HSIF023S PROCESSING TERMINATED - HSIINQZP *func* FAILURE, RC=*rc*

Explanation: HSIINQZP file processing failure.

In the message text:

func
file processing in error.

rc return code.

System action: Terminates with a condition code of 16.

Operator response: None.

System programmer response: Examine the return code shown in the message and use for problem determination/resolution.

Module: HSIIS2D

HSIF024S PROCESSING TERMINATED - HSIINQOT *func* FAILURE, RC=*rc*

Explanation: HSIINQOT file processing failure.

In the message text:

func
file processing in error.

rc return code.

System action: Terminates with a condition code of 16.

Operator response: None.

System programmer response: Examine the return code shown in the message and use for problem determination/resolution.

Module: HSIIS2D

HSIF025S PROCESSING TERMINATED - TABLE #LPRTBL# *func* FAILURE, RC=*rc*

Explanation: Table processing failure.

In the message text:

func
table operation in error.

rc return code.

System action: Terminates with a condition code of 16.

Operator response: None.

System programmer response: Examine the return code shown in the message and use for problem determination/resolution.

Module: HSIIS2D

HSIF999U HSIMSG/HSIFMSG FAILURE - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Ensure Joblib/Steplib contains the library where the HSIFMSG message module resides. If you cannot resolve this issue then contact IBM support.

Module: HSIIM2D HSIIS2D :emsgl.

HSII - Utility messages

Return codes

0	No errors encountered. All requests processed successfully.
4	I/O error in one or more program libraries.
8	Error - Incomplete data. Processing continues. OPEN or other system service error.
12	Syntax error.
16	Unrecoverable error.
20	Disastrous error. No requests processed. SYSPRINT file cannot be used.

Message suffix codes

Suffix	Meaning	Raises Minimum Condition Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSII001I READ FAILED FOR SYSIN, RC=*rc*

Explanation: The HSIICUST program could not read commands from the SYSIN DDname.

In the message text:

rc return code from EXECIO

System action: The program terminates and takes no actions.

Operator response: Correct the JCL and provide a SYSIN DD with valid control statements.

System programmer response: None.

Module: HSIICUST

HSII002I REQUIRED PARAMETER *prm* IS MISSING FROM SYSIN

Explanation: The HSIICUST program did not find the required parm in the SYSIN supplied by the user.

In the message text:

prm
name of the parm that is missing.

System action: The program terminates and takes no actions.

Operator response: Correct the SYSIN and supply the required parm.

System programmer response: None.

Module: HSIICUST

HSII003I THE DEFAULT VALUE "*prm*=*dft*" IS BEING USED.

Explanation: The submitted HSISCUST Job SYSIN did not contain this parameter and the default value will now be used.

In the message text:

prm
name of the parameter.

dft
supplied default.

System action: The program continues and uses the default as documented in the message.

Operator response: If the default value is to be overridden supply the parameter value in the HSISCUST Job SYSIN stream then resubmit.

System programmer response: None.

Module: HSIICUST

HSII004I ALLOCATION OF DATASET *dsn* TO *dd* FAILED, RC=*rc*

Explanation: The HSIICUST program could not allocate the dataset to the ddname.

In the message text:

dsn
name of the dataset that failed allocation.

dd DD name to be allocated.

rc return code from the allocate command.

System action: The program terminates and takes no actions.

Operator response: Check the return code from the allocate command in the TSO commands manual. Correct the options and try running the program again. The probable option in error is HSIINST.

System programmer response: None.

Module: HSIICUST

HSII005I *func* FAILED FOR *rsc*, RC=*rc*

Explanation: The HSIICUST program could not perform a required ISPF function because an error occurred during the ISPF function execution.

In the message text:

func
name of ISPF function that failed.

rsc
resource that caused the failure.

rc return code from the ISPF function.

System action: The program terminates and takes no actions. The program may have written out JCL and parameter members. These members should be treated as suspect as they might contain errors in them due to the nature of this error.

Operator response: Check that the options specified do not exceed the field length requirements of the various options. If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIICUST

HSII006I MODEL DATASET SHSISAMP HAS NOT BEEN FOUND.

Explanation: The HSIICUST program could not find the SHSISAMP dataset created during installation.

System action: The program terminates and takes no actions.

Operator response: Check that the HSI option is correctly specified and that the installation target libraries are available to the customization program.

System programmer response: None.

Module: HSIICUST

HSII007I CUSTOMIZATION TERMINATES ...

Explanation: The HSIICUST program encountered an error during execution.

System action: The program terminates and takes no further action.

Operator response: Check the previous message which will identify the component causing the problem.

System programmer response: None.

Module: HSIICUST

HSII008I *prm* PARAMETER VALUE *pval* IS > *plen* CHARACTERS.

Explanation: The HSIICUST program found a parameter value with a length greater than the allowed value for that parm.

In the message text:

prm
name of HSISCUST parameter that failed.

pval
contents of the PARM.

plen
length of Parameter value.

System action: The program terminates and takes no actions.

Operator response: Check the parameter in question and re-submit the HSISCUST JCL after correcting the length error.

System programmer response: None.

Module: HSIICUST

HSII009I *prm* **DATASET** *dsn* **WAS NOT FOUND ON SYSTEM** *systm*

Explanation: The HSIICUST program found a parameter value, which requires a Dataset name. This Dataset name could not be found on the system.

In the message text:

prm
name of HSISCUST parameter that failed.

dsn
dataset name associated with the PARM.

systm
Operating System name.

System action: The program terminates and takes no actions.

Operator response: Check the parameter in question and re-submit the HSISCUST JCL after correcting the DSN error.

System programmer response: None.

Module: HSIICUST

HSII010I *verb* **STATEMENT VERB NOT ONE OF** *list*

Explanation: During syntax parsing for a statement the verb found does not match any of the valid verbs for the program.

In the message text:

verb
word that is not a valid verb.

list
list of valid verbs.

System action: The program terminates and takes no actions.

Operator response: Update the statements to the program to correct the verb in error and supply a correct verb.

System programmer response: None.

Module: HSIIREGN

HSII011I *word* **NOT VALID FOR VERB** *verb*

Explanation: During syntax parsing for a statement, a word was found that does not match the syntax of the statement for the verb that is being processed.

In the message text:

word
word that is not valid for the statement syntax for a verb.

verb
the verb of the statement that encountered the error.

System action: The program terminates and takes no actions.

Operator response: Update the statements to the program to correct the statement in error.

System programmer response: None.

Module: HSIIREGN

HSII012I **PARAMETER** *prm* **IS NOT A VALID HSISCUST PARAMETER**

Explanation: An invalid HSISCUST SYSIN parameter has been found.

In the message text:

prm
parameter that is invalid.

System action: The program terminates.

Operator response: Remove the invalid parameter from the HSISCUST Job SYSIN then resubmit.

System programmer response: None.

Module: HSIICUST

HSII013I **PARAMETER** *prm* **HAS A NULL VALUE**

Explanation: The submitted HSISCUST Job SYSIN has encountered a parameter with a NULL value.

In the message text:

prm
name of the parameter that is null.

System action: The program terminates.

Operator response: Ensure that the HSISCUST parameter has a valid parameter value in the HSISCUST Job SYSIN then resubmit.

System programmer response: None.

Module: HSIICUST

HSII014I PARAMETER VALUE FOR *prm* HAS UNBALANCED QUOTES, PARAMETER VALUE IS *pval*

Explanation: An HSISCUST parameter contains unbalanced quotes.

In the message text:

prm
name of the parameter with unbalanced quotes.

pval
parameter value.

System action: The program terminates

Operator response: Ensure that the Parameter value is enclosed within single quotation marks then resubmit the HSISCUST Job.

System programmer response: None.

Module: HSIICUST

HSII015I DATASET *dsn stat*

Explanation: The HSIICUST program identifies the status of the output datasets that it is going to use.

In the message text:

dsn
name of output dataset.

stat
status of the output dataset.

System action: The HSIICUST program continues processing.

Operator response: Informational message, no action required.

System programmer response: None.

Module: HSIICUST

HSII016I UNMATCHED COMMENT DELIMITER IN HSISCUST STATEMENT: *stmnt*

Explanation: The HSIICUST program found an error in the comment delimiters passed from the HSISCUST SYSIN stream.

In the message text:

stmnt
statement where the error occurred.

System action: The program terminates and takes no actions.

Operator response: Correct the HSISCUST SYSIN statements and provide valid comment delimiters, /* */.

System programmer response: None.

Module: HSIICUST

HSII017I PARAMETER *prm* MUST HAVE THE 1ST CHARACTER AS A VALUE BETWEEN A AND Z

Explanation: The HSISCUST Job has encountered a parameter in the SYSIN DD stream where the 1st character is not alphabetic. This parameter value must start with a value between A and Z.

In the message text:

prm
name of the parameter that has a non alphabetic 1st character.

System action: The program terminates.

Operator response: Ensure that the HSISCUST parameter has a value between A and Z then resubmit the Job.

System programmer response: None.

Module: HSIICUST

HSII018I *prm* VALUE *pval* MUST BE WITHIN THE VALID RANGE OF *val1* TO *val2*

Explanation: The HSIICUST program encountered a parameter that was outside the valid range of values allowed.

In the message text:

prm
name of the parameter in error.

pval
value of parameter in error.

val1
lower limit value of valid range.

val2
upper limit value of valid range.

System action: The program terminates and takes no actions.

Operator response: Correct the parameter in error and rerun the HSISCUST Job.

System programmer response: None.

Module: HSIICUST

HSII019I PARAMETER *prm* MUST BE EITHER "Y" OR "N", PARAMETER VALUE IS *pval*

Explanation: The HSISCUST parameter contains a value other than "Y" or "N".

In the message text:

prm
name of the parameter.

pval
parameter value.

System action: The program terminates

Operator response: Ensure that the Parameter value is equal to "Y" or "N" as required then re-submit the HSISCUST Job.

System programmer response: None.

Module: HSIICUST

HSII020I TAILORING PARAMETERS:

Explanation: HSIICUST progress message.

System action: The program continues.

Operator response: This is an informational progress message and no further action is required.

System programmer response: None.

Module: HSIICUST

HSII021I CREATING POST-INSTALLATION DATASETS:

Explanation: HSIICUST progress message.

System action: The program continues.

Operator response: This is an informational progress message and no further action is required.

System programmer response: None.

Module: HSIICUST

HSII022I APPLYING TAILORING INFORMATION TO POST-INSTALLATION MEMBERS:

Explanation: HSIICUST progress message.

System action: The program continues.

Operator response: This is an informational progress message and no further action is required.

System programmer response: None.

Module: HSIICUST

HSII023I POST-INSTALLATION CUSTOMIZATION COMPLETE.

Explanation: HSIICUST completion message.

System action: The program ends successfully.

Operator response: This is an informational progress message and no further action is required.

System programmer response: None.

Module: HSIICUST

HSII024I PARAMETER *prm* CONTAINS AN EXTRANEOUS VALUE: *xval*

Explanation: An HSISCUST parameter contains an extraneous value.

In the message text:

prm
name of the parameter with an extraneous value.

xval
the extraneous value(s).

System action: The program terminates

Operator response: Ensure that the Parameter value is correctly defined and that any comments are enclosed within comment delimiters.

System programmer response: None.

Module: HSIICUST

HSII025I PARAMETER VALUE FOR *prm* MUST BE IN QUOTES, PARAMETER VALUE IS *pval*

Explanation: An HSISCUST parameter value contains no quotes.

In the message text:

prm
name of the parameter.

pval
parameter value.

System action: The program terminates

Operator response: Ensure that the Parameter value is enclosed within single quotation marks then resubmit the HSISCUST Job.

System programmer response: None.

Module: HSIICUST

HSII026I PARAMETER *prm* MUST BE A VALUE BETWEEN A-Z OR 0-9

Explanation: The HSISCUST Job has encountered a parameter in the SYSIN DD stream where the 1st character is not alphanumeric. This parameter value must be a value between A-Z or 0-9

In the message text:

prm
name of the parameter that has a non alphanumeric first character.

System action: The program terminates.

Operator response: Ensure that the HSISCUST parameter has a value between A-Z or 0-9 then resubmit the Job.

System programmer response: None.

Module: HSIICUST

HSII100I *prm* **MISSING FROM CONFIGURATION.**

Explanation: The Region Assignment utility requires the parameter in the TPARAM dataset.

In the message text:

prm
parameter that is missing.

System action: The program terminates and takes no actions.

Operator response: Update the parameters in the TPARAM DD to add the missing parameter.

System programmer response: None.

Module: HSIIREGN

HSII101I **ATTEMPT TO DEFINE REGION** *regid* **FAILED, REGION ALREADY EXISTS.**

Explanation: The DEFINE REGION statement contains a region name that already exists.

In the message text:

regid
region to be defined.

System action: The program does not process this statement. Statements following the statement in error are not processed.

Operator response: Change the DEFINE REGION statement to refer to a region name that does not already exist.

System programmer response: None.

Module: HSIIREGN

HSII102I **REGION** *regid* **HAS NOT BEEN DEFINED.**

Explanation: The ASSIGN REGION statement contains a region name that does not exist.

In the message text:

regid
region to be assigned.

System action: The program does not process this statement. Statements following the statement in error are not processed.

Operator response: Change the ASSIGN REGION statement to refer to a region name that exists.

System programmer response: None.

Module: HSIIREGN

HSII103I **INVENTORY** *inv* **HAS NOT BEEN DEFINED.**

Explanation: The ASSIGN INVENTORY statement contains an inventory that does not exist.

In the message text:

inv
inventory to be assigned.

System action: The program does not process this statement. Statements following the statement in error are not processed.

Operator response: Change the ASSIGN INVENTORY statement to refer to an inventory name that exists.

System programmer response: None.

Module: HSIIREGN

HSII104I **REGION** *regid* **IS ALREADY ASSIGNED TO** *prg*

Explanation: The ASSIGN REGION statement contains a region name that is to be assigned to another region, but the region being assigned is already assigned to a region.

In the message text:

regid
region to be assigned.

prg
parent region name.

System action: The program does not process this statement. Statements following the statement in error are not processed.

Operator response: Do not attempt to assign regions that are already assigned.

System programmer response: None.

Module: HSIIREGN

HSII105I **REGION** *regid* **DEFINED.**

Explanation: The DEFINE REGION statement has executed correctly.

In the message text:

regid
region to be defined.

System action: The program continues with processing.

Operator response: None.

System programmer response: None.

Module: HSIIREGN

HSII106I INVENTORY *inv* ASSIGNED TO *regid*

Explanation: The ASSIGN INVENTORY statement has executed correctly.

In the message text:

inv
inventory to be assigned.

regid
region name.

System action: The program continues with processing.

Operator response: None.

System programmer response: None.

Module: HSIIREGN

HSII107I *svc* FROM *rsc* FAILED, RC=*rc*

Explanation: An error has occurred executing the service for the resource specified. The service issued the return code mentioned.

In the message text:

svc
service that failed.

rsc
resource that failed.

rc return code from service.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Report this error to the System Programmer.

System programmer response: For "EXECIO READ" service, this probably means that the resource specified (ddname) is missing or empty. If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIREGN

HSII108I SQL *verb* FAILED, SQLCODE=*sqlcode*

Explanation: An error has occurred executing the SQL verb for the table specified.

In the message text:

verb
SQL verb and table name.

sqlcode
SQLCODE from failing statement.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Report this error to the System Programmer.

System programmer response: If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIREGN

HSII109I INVENTORY *inv* IS ALREADY ASSIGNED TO *regid*

Explanation: An error has occurred executing the SQL verb for the table specified.

In the message text:

inv
inventory name.

regid
region name.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Report this error to the System Programmer.

System programmer response: If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIREGN

HSII110I INVENTORY *inv* REMOVED FROM *regid*

Explanation: The inventory specified has been successfully removed from the region specified.

In the message text:

inv
inventory name.

regid
region name.

System action: The program continues processing.

Operator response: None.

System programmer response: None.

Module: HSIIREGN

HSII111I INVENTORY *inv* FAILED REMOVAL FROM *regid*, SQLCODE=*sqlcd*

Explanation: The inventory specified has not been removed from the region specified.

In the message text:

inv
Inventory name.

regid
Region name.

sqlcd
SQL Return Code.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Report this error to the System Programmer.

System programmer response: If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIREGN

HSII112I CANNOT DELETE REGION *regid* **-** *regnm*

Explanation: The region specified cannot be deleted from the database.

In the message text:

regid
region ID.

regnm
region name.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Do not delete the region specified.

System programmer response: None.

Module: HSIIREGN

HSII113I CANNOT DELETE REGION *regnm*, **REGION HAS CHILDREN.**

Explanation: The region specified cannot be deleted because it has other regions assigned to it.

In the message text:

regnm
region name.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Remove the region assignments from the region, before trying to delete the region.

System programmer response: None.

Module: HSIIREGN

HSII114I CANNOT DELETE REGION *regnm*, **REGION HAS INVENTORIES ASSIGNED.**

Explanation: The region specified cannot be deleted because it has inventories assigned to it.

In the message text:

regnm
region name.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Remove the inventory assignments from the region, before trying to delete the region.

System programmer response: None.

Module: HSIIREGN

HSII115I REGION *regnm* **DELETED.**

Explanation: The region specified has been deleted.

In the message text:

regnm
region name.

System action: The program continues processing.

Operator response: None.

System programmer response: None.

Module: HSIIREGN

HSII116I REGION *regnm* **NOT DELETED, SQLCODE=***sqlcode*

Explanation: The region specified was not deleted due to an SQL error. SQLCODE details the error that occurred.

In the message text:

regnm
region name.

sqlcode
SQL Return Code.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Report this error to the System Programmer.

System programmer response: If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIREGN

HSII117I REGION *regnm* ASSIGNED TO *prg*

Explanation: The region specified was assigned to a parent region.

In the message text:

regnm
region name.

prg
parent region name.

System action: The program continues processing.

Operator response: None.

System programmer response: None.

Module: HSIIREGN

HSII118I REGION *regnm* CONTAINS DBCS CHARACTERS.

Explanation: The region name specified must not contain DBCS characters.

In the message text:

regnm
region name.

System action: The program stops processing statements. The current statement changes to DB2 tables are backed out.

Operator response: Do not put DBCS characters in the region name.

System programmer response: None.

Module: HSIIREGN

HSII200I *ddn* DDNAME CSVIN IS MISSING.

Explanation: The SCRT Import utility requires the DDname CSVIN

In the message text:

ddn
DDname that is missing.

System action: The program terminates and takes no actions.

Operator response: Add the CSVIN DDname to the JCL and include the required input dataset containing data from the IBM SCRT tool.

System programmer response: None.

Module: HSIISCRT

HSII201I *dsn* CSVIN DATASET IS NOT PO OR PS.

Explanation: SCRT Import Utility requires the CSVIN dsn to have a DSORG of PO or PS

In the message text:

dsn
CSVIN dataset not PO or PS.

System action: The program terminates and takes no actions.

Operator response: Ensure that the CSVIN dsn has a DSORG of PO or PS before submitting the Job.

System programmer response: None.

Module: HSIISCRT

HSII202I MISSING PID *pid*.

Explanation: The SCRT Import utility has detected a missing PID.

In the message text:

pid
missing PID.

System action: The program continues.

Operator response: Supply IBM support with the missing PID number and request a Global Knowledge Base (GKB) refresh.

System programmer response: None.

Module: HSIISCRT

HSII300I ERROR WRITING TO *ddn*.

Explanation: The XML Export utility has a problem writing the XML file.

In the message text:

ddn
DDNAME of the file

System action: The program terminates.

Operator response: Check the Return Code and any preceding messages.

System programmer response: None.

Module: HSIIKBT

HSII301I NUMBER OF LINES WRITTEN TO SWKBTXML DD IS *ocnt*.

Explanation: No. of Lines written to SWKBTXML DD by the XML Export utility.

In the message text:

ocnt
lines written to SWKBTXML DD.

System action: The program continues and takes no actions.

Operator response: Informational message, no action required.

System programmer response: None.

Module: HSIIKBT

HSII302I SQL WARNING FOR *warn*.

Explanation: SQL warning was issued from a command

In the message text:

warn
SQL warning.

System action: The program continues.

Operator response: None.

System programmer response: None.

Module: HSIIKBT

HSII303I SQL ERROR FOR *err*.

Explanation: The XML Export Utility has encountered an error.

In the message text:

err
SQL Error.

System action: The program terminates and takes no actions.

Operator response: Examine the SQL return code to determine the cause of the error. Inform the system programmer.

System programmer response: If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIKBT

HSII304I *err*.

Explanation: The XML Export Utility has encountered a DSNREXX error.

In the message text:

err
DSNREXX error.

System action: The program terminates and takes no actions.

Operator response: Examine the preceding error messages to determine the error. Inform the system programmer.

System programmer response: If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIIKBT

HSII305I INVALID SCHEMA *sch*.

Explanation: The XML Export Utility has encountered a problem with an invalid Schema.

In the message text:

sch
schema name

System action: The program terminates and takes no actions.

Operator response: Ensure that the correct Schema is being used.

System programmer response: None.

Module: HSIIKBT

HSII999U MODULE HSIIMSG FAILED - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Contact IBM Support.

Module: HSIIINQ :emsgl.

HSIP - Inquisitor for z/OS messages and codes

Return codes

0	No errors encountered. All requests processed successfully.
4	I/O error in one or more program libraries.
8	Error - Data collection is incomplete. Processing continues. The error is in a system service such as OPEN or DYNALLOC. Data that is collected can be processed normally.
12	Syntax error.
16	Unrecoverable error. No requests processed. SYSIN or HSIPZIP or HSIPOUT File cannot be used, or unsupported Operating System.
20	Disastrous error. No requests processed. SYSPRINT file cannot be used.

Message suffix codes

Suffix	Meaning	Raises Minimum Condition Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIP000U NO USABLE SYSPRINT FILE

Explanation: The OPEN of the SYSPRINT file failed. Note: This message is issued by WTO with ROUTCDE=(2,11). All other messages are written to the SYSPRINT file.

System action: Terminates with a condition code of 20.

Operator response: Ensure a usable SYSPRINT file is allocated. The program overrides any unacceptable DCB values.

System programmer response: None.

Module: HSIPINQ

HSIP001U CANNOT OPEN SYSIN FILE

Explanation: The OPEN of the SYSIN file failed.

System action: Terminates with a condition code of 16.

Operator response: Ensure a usable SYSIN file is allocated.

System programmer response: None.

Module: HSIPINQ

HSIP004S UNKNOWN VERB "*verb*"

Explanation: Parsing detected unrecognised data when looking for a verb.

In the message text:

verb
name of the encountered verb.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP005S UNKNOWN OPERAND "*op*"

Explanation: Parsing detected unrecognised data when looking for an operand.

In the message text:

op name of the encountered operand.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP006S UNEXPECTED LEFT PARENTHESIS ENCOUNTERED

Explanation: Parsing detected a left parenthesis at an unexpected location.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP007S UNEXPECTED RIGHT PARENTHESIS ENCOUNTERED

Explanation: Parsing detected a right parenthesis at an unexpected location.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP008S EXPECTED LEFT PARENTHESIS MISSING

Explanation: Parsing did not detect the expected left parenthesis.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP009S EXPECTED RIGHT PARENTHESIS MISSING

Explanation: Parsing did not detect the expected right parenthesis.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP010U OPERATING SYSTEM NOT SUPPORTED - CODE "*code*"

Explanation: The value of the byte at CVTDCB was not X'9B'.

In the message text:

code
hexadecimal value of first byte of CVTDCB.

System action: Terminates with a condition code of 16.

Operator response: This version of the Inquisitor cannot be run on this Operating System. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP011I MISSING RIGHT PARENTHESIS ASSUMED

Explanation: End-of-file was detected for SYSIN before an expected right parenthesis was detected.

System action: The request is accepted and processing continues.

Operator response: Correct the SYSIN file contents to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP012S MISSING OPERAND SUBPARAMETER FOR *spm*

Explanation: A required subparameter of an operand was not specified.

In the message text:

spm
name of the operand being processed.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP013S E-O-F INSTEAD OF EXPECTED CONTINUATION

Explanation: End-of-file was detected for SYSIN instead of an expected record required to continue the current statement being parsed.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP014I COMPLETED REQUEST NUMBER *rno* - PROCESSING STATISTICS ARE:

Explanation: Processing of a request has been completed. A HSIP015I message follows containing the statistics for the request.

In the message text:

rno
sequence number of the request.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIPINQ

HSIP015I VOLUMES=*vol* DATASETS=*ds* BAD-D/S=*dsbad* PROGRAMS=*pgms*

Explanation: Processing of a request has been completed. Statistics related to the request are shown.

In the message text:

vol
count of volumes scanned for this request.

ds count of data sets successfully processed.

dsbad
count of data sets which could not be processed.

pgms
count of programs processed for this request.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIPINQ

HSIP016I ACCEPTED REQUEST NUMBER *rno*

Explanation: Parsing of a request has been completed successfully. The request is stored for subsequent processing.

In the message text:

rno
sequence number of the request.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP017E DYNALLOC FAILURE: RC=*rc* ERROR=*err* INFO=*inf* VOLUME=*vol*

Explanation: A data set could not be dynamically allocated. See message HSIP080I for the name of the dataset that incurred the problem.

In the message text:

rc return code of the DYNALLOC macro.

err
dynamic allocation return code (DARC).

inf
dynamic allocation information code.

vol
volume serial number of the data set.

System action: Processing of this data set is terminated.

Operator response: If necessary, rerun when the file is available for use. Note: The meanings of many DARC values are usually available in Appendix A of the ISPF Tutorial.

System programmer response: None.

Module: HSIPINQ

HSIP018E VTOC DYNALLOC FAILURE: RC=*rc* ERROR=*err* INFO=*inf* VOLUME=*vol*

Explanation: A VTOC could not be dynamically allocated.

In the message text:

rc return code of the DYNALLOC

err
dynamic allocation return code (DARC).

inf
dynamic allocation information code.

vol
volume serial number of the data set.

System action: Processing of this volume is terminated.

Operator response: If necessary, rerun when the VTOC is available for use to process this volume. Note: The meanings of many DARC values are usually available in Appendix A of the ISPF Tutorial.

System programmer response: None.

Module: HSIPINQ

HSIP020I *ocnt* **INQUISITOR OUTPUT RECORDS WRITTEN**

Explanation: Processing has concluded and all data files have been closed.

In the message text:

ocnt
number of records written.

System action: Termination continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP021S **INVALID OPERAND SUBPARAMETER FOR** *spm*

Explanation: The specified subparameter of an operand was not valid.

In the message text:

spm
name of the operand being processed.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP022W **I/O ERR MEMBER** *mbr* **IN** *dsn*

Explanation: An I/O error was encountered while reading the contents of a load module.

In the message text:

mbr
name of the program being processed.

dsn
name of the data set being processed.

System action: Processing of this member continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP023E **ABEND** *abend* **IN OPEN FOR** *dsn*

Explanation: An abnormal end occurred while opening a data set.

In the message text:

abend
hexadecimal system abend and reason

dsn
name of the data set being processed.

System action: Processing of this data set is terminated.

Operator response: None required, but you may wish to exclude the data set from processing, or correct the cause of the abend.

System programmer response: None.

Module: HSIPINQ

HSIP024S **BAD UCBSCAN RETURN CODE OF HEX** *rc*

Explanation: An unexpected return code was received from UCBSCAN.

In the message text:

rc hexadecimal return code from UCBSCAN

System action: Processing of volume scanning for this request is terminated.

Operator response: Rerun the program when no dynamic reconfiguration changes are being implemented.

System programmer response: None.

Module: HSIPINQ

HSIP025U **CANNOT OPEN HSIPOUT FILE**

Explanation: The OPEN of the HSIPOUT file failed.

System action: Terminates with a condition code of 16.

Operator response: Ensure that the allocated HSIPOUT file is usable, or omit the HSIPOUT file in favour of using the HSIPZIP file.

System programmer response: None.

Module: HSIPINQ

HSIP026E **I/O ERROR ENCOUNTERED READING VTOC OF VOLUME** *vol* **ON DEVICE** *dev*

Explanation: An I/O error was encountered while reading a VTOC.

In the message text:

vol
volume serial number being processed.

dev
device number of the volume.

System action: Processing of this track of the VTOC is terminated.

Operator response: None required, but you may wish

to exclude the volume from processing, or correct the cause of the I/O error.

System programmer response: None.

Module: HSIPINQ

HSIP028U CANNOT OPEN HSIPDMP FILE

Explanation: The OPEN of the HSIPDMP file failed after DUMPTEXT was specified.

System action: Terminates with a condition code of 16.

Operator response: Ensure a usable HSIPDMP file is allocated, or remove all DUMPTEXT operand's from the contents of the SYSIN file. The DUMPTEXT operand should only be specified at the request of IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP029I TEXT-DUMPS=*cnt*

Explanation: Processing of a request with DUMPTEXT specified has completed. This message follows HSIP015I.

In the message text:

cnt
count of load module text blocks written.

System action: Processing continues.

Operator response: None required. The DUMPTEXT operand should only be specified at the request of IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP030I "DUMPTEXT" OPERAND IGNORED FOR "SCANDIR" VERB

Explanation: A DUMPTEXT operand was encountered for a SCANDIR request. That is, the possible dumping of load module text blocks was specified in a request which does not have access to text blocks.

System action: The DUMPTEXT operand is ignored and processing continues.

Operator response: Remove the DUMPTEXT operand to avoid this message. The DUMPTEXT operand should only be specified at the request of IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP031I BAD SELECTION CRITERIA FOR *dsn*

Explanation: Processing of a data set was specified but attributes did not match other selection criteria also specified in the request. This message is followed by HSIP038I which details the cause.

In the message text:

dsn
name of the data set being processed.

System action: Processing of this data set is terminated.

Operator response: If this data set is a program library which should be processed by the Inquisitor then modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP032I OBTAIN FAILED RC=*rc* VOLUME *vol*

Explanation: The system could not read the VTOC entry for the data set named in the HSIP033I message which follows this message. This message is only issued when a program parameter of "DSNMSG" or "ALLMSG" is specified.

In the message text:

rc hexadecimal return code of the OBTAIN macro.

vol
volume serial number being processed.

System action: Processing of this data set is terminated.

Operator response: Ensure the relevant catalog entry is correct. Ensure the relevant volume is online and available to the system. Ensure there is no I/O error in the relevant volume's VTOC. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP033I OBTAIN FAILED FOR DATA SET *dsn*

Explanation: The system could not read the VTOC entry for the data set on the volume named in the previous HSIP032I message. This message is only issued when a program parameter of "DSNMSG" or "ALLMSG" is specified.

In the message text:

dsn
name of the data set being processed.

System action: Processing of this data set is terminated.

Operator response: Ensure the relevant catalog entry

is correct. Ensure the relevant volume is online and available to the system. Ensure there is no I/O error in the relevant volumes VTOC. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP034I REFER DATE WAS *date* FOR *dsn*

Explanation: A load library was opened. The reference date of the data set before the OPEN is reported in this message. This message is only issued when a program parameter of "DSNMSG" or "ALLMSG" is specified.

In the message text:

date
the Julian reference date from the VTOC entry.

dsn
name of the data set being processed.

System action: Processing of this data set continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP036E OPEN ERROR ENCOUNTERED READING VTOC OF VOLUME *vol* ON DEVICE *dev*

Explanation: The VTOC of the volume shown could not be opened.

In the message text:

vol
volume serial number being processed.

dev
device number of the volume.

System action: Processing of this track of the VTOC is terminated.

Operator response: None required, but you may wish to exclude the volume from processing, or correct the cause of the I/O error.

System programmer response: None.

Module: HSIPINQ

HSIP037E SECURITY ACCESS DENIED FOR *dsn*

Explanation: A RACROUTE macro determined the program had insufficient security access to read the data set.

In the message text:

dsn
name of the data set being processed.

System action: Processing of this data set is terminated.

Operator response: Contact Security Administration to obtain sufficient security access to read the data set or exclude the data set from processing.

System programmer response: None.

Module: HSIPINQ

HSIP038I BAD SELECTION CRITERIA WAS *dsn*

Explanation: Processing of a data set was specified but attributes did not match other selection criteria also specified in the request. This message follows HSIP031I which shows the data set name.

In the message text:

dsn
cause of the data set processing failure.

System action: Processing of this data set is terminated.

Operator response: If this data set is a program library which should be processed by the Inquisitor then modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP039S ALL POSSIBLE DATA SETS ARE EXCLUDED

Explanation: An exclusion mask has been specified which excludes all possible data sets included by a selection mask. Both masks are shown after this message.

System action: Terminates with a condition code of 12.

Operator response: Modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP040S ALL POSSIBLE DASD VOLUMES ARE EXCLUDED

Explanation: An exclusion mask has been specified which excludes all possible DASD volumes included by a selection mask. Both masks are shown after this message.

System action: Terminates with a condition code of 12.

Operator response: Modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP041S ALL POSSIBLE PROGRAMS ARE EXCLUDED

Explanation: An exclusion mask has been specified which excludes all possible programs included by a selection mask. Both masks are shown after this message.

System action: Terminates with a condition code of 12.

Operator response: Modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP042S ALL POSSIBLE MODULES ARE EXCLUDED

Explanation: An exclusion mask has been specified which excludes all possible modules included by a selection mask.

System action: Terminates with a condition code of 12.

Operator response: Modify or remove the conflicting selection criteria. If no CSECT-level records are required then omit both MODULE and XMODULE operands.

System programmer response: None.

Module: HSIPINQ

HSIP043I "MODULE"/"CSECT" OPERAND IGNORED FOR "SCANDIR" VERB

Explanation: A MODULE operand was encountered for a SCANDIR request. That is, the output of program structure data was requested in a request which does not have access to this data.

System action: The MODULE operand is ignored and processing continues.

Operator response: Remove the MODULE operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP044I "XMODULE"/"XCSECT" OPERAND IGNORED FOR "SCANDIR" VERB

Explanation: An XMODULE operand was encountered for a SCANDIR request. That is, the output of program structure data was implied in a request which does not have access to this data.

System action: The XMODULE operand is ignored and processing continues.

Operator response: Remove the XMODULE operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP045I THE "XDSNAME" MASK IS NOT A SUBSET OF ANY "DSNAME" MASK

Explanation: The mask specified in the XDSNAME operand excludes possible values not included in the DSNAME operand. This message is issued to highlight possible inconsistencies in a request.

System action: Processing continues.

Operator response: Specify the XDSNAME operand as a further qualification of the DSNAME operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP046I THE "XVOLUME" MASK IS NOT A SUBSET OF ANY "VOLUME" MASK

Explanation: The mask specified in the XVOLUME operand excludes possible values not included in the VOLUME operand. This message is issued to highlight possible inconsistencies in a request.

System action: Processing continues.

Operator response: Specify the XVOLUME operand as a further qualification of the VOLUME operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP047I THE "XPROGRAM" MASK IS NOT A SUBSET OF ANY "PROGRAM" MASK

Explanation: The mask specified in the XPROGRAM operand excludes possible values not included in the PROGRAM operand. This message is issued to highlight possible inconsistencies in a request.

System action: Processing continues.

Operator response: Specify the XPROGRAM operand as a further qualification of the PROGRAM operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP048I THE "XMODULE" MASK IS NOT A SUBSET OF ANY "MODULE" MASK

Explanation: The mask specified in the XMODULE operand excludes possible values not included in the MODULE operand. This message is issued to highlight possible inconsistencies in a request.

System action: Processing continues.

Operator response: Specify the XMODULE operand as a further qualification of the MODULE operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP049I THE "XSTOGROUP" MASK IS NOT A SUBSET OF ANY "STOGROUP" MASK

Explanation: The mask specified in the XSTOGROUP operand excludes possible values not included in the STOGROUP operand. This message is issued to highlight possible inconsistencies in a request.

System action: Processing continues.

Operator response: Specify the XSTOGROUP operand as a further qualification of the STOGROUP operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP050I MODULES=*cnt*

Explanation: Processing of a request with MODULE specified has completed. This message follows HSIP015I.

In the message text:

cnt
count of CSECTs processed in this request.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP051I ****** PARSE ONLY REQUEST PROCESSED - NO ACTION TAKEN ********

Explanation: Processing of a SCANCMD request has completed.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP052U MISSING HSIPOUT AND HSIPZIP FILES

Explanation: Neither an HSIPOUT nor an HSIPZIP file is allocated. At least one output file is required.

System action: Terminates with a condition code of 16.

Operator response: Specify an output file and rerun the job.

System programmer response: None.

Module: HSIPINQ

HSIP053U COMPRESSION SUBROUTINE ERROR

Explanation: While processing the HSIPZIP file the compression subroutine encountered an error. The error message from the compression subroutine immediately follows this message.

System action: Terminates with a condition code of 16.

Operator response: Correct the error described in the message from the compression subroutine. If necessary, gather appropriate diagnostic materials and contact IBM support

System programmer response: None.

Module: HSIPINQ

HSIP054I "FULLDIR" OPERAND IGNORED FOR "SCANDIR" VERB

Explanation: A FULLDIR operand was encountered for a SCANDIR request. That is, the processing of load module member data was specified in a request which does not have access to this data.

System action: The FULLDIR operand is ignored and processing continues.

Operator response: Remove the FULLDIR operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP056I *date time* COMMENCING SCAN OF VOLUME *vol* ON UNIT *unit*

Explanation: A request without the CATALOG keyword began processing a DASD volume. This message provides feedback on the progress of long-running Inquisitor requests.

In the message text:

date
date of message.

time
time of message.

vol
serial number of volume.

unit
device number of volume.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP057E ABEND *abend* IN OPEN FOR VTOC OF VOLUME *vol* ON UNIT *unit*

Explanation: A request without the CATALOG keyword attempted to open a DASD volume VTOC and the OPEN abended. The volume is not usable.

In the message text:

abend
hexadecimal system abend and reason codes.

vol
serial number of volume.

unit
device number of volume being processed.

System action: Processing of this volume is terminated.

Operator response: Vary the volume offline, and/or reformat the volume. Institute any appropriate volume recovery procedures.

System programmer response: None.

Module: HSIPINQ

HSIP058S DUPLICATE OPERAND ENCOUNTERED: *op*

Explanation: An input request was found to have the indicated operand specified more than once.

In the message text:

op name of the duplicate operand

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program.

System programmer response: None.

Module: HSIPINQ

HSIP059E BINDER FAILURE FOR MEMBER *mbr* RC=*rc* RS=*rs*

Explanation: The Binder could not successfully process a member of a PDSE.

In the message text:

mbr
name of the member being processed.

rc hexadecimal Binder FDA API return code.

rs hexadecimal Binder FDA API reason code.

System action: Terminates data collection for this member, writes out data already collected and continues processing the next member.

Operator response: None required.

System programmer response: The Binder Fast Data Access API return and reason codes provide more detailed indication of the cause.

Module: HSIPINQ

HSIP060S SYMBOL SUBSTITUTION FAILURE - ASASYMBP RC=*rc*

Explanation: The system symbol substitution routine could not successfully perform symbol substitution. Data before and after substitution is shown in the SYSPRINT file.

In the message text:

rc hexadecimal return code.

System action: Terminates with a condition code of 12.

Operator response: Correct or remove the symbols in control statement input.

System programmer response: None.

Module: HSIPINQ

HSIP061I *pgm* NON-REEDITABLE IN *dsn*

Explanation: A program object in a PDSE was encountered which cannot be processed by the Program Binder. This message is only issued when a program parameter of "PGMMSG" or "ALLMSG" is specified.

In the message text:

pgm
name of program which cannot be processed.

dsn
name of the data set being processed.

System action: Further data collection for this member is terminated.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP062S THE CATALOG REQUEST NEEDS EXACTLY ONE DSNAME MASK

Explanation: A request with the CATALOG operand either omitted the DSNAME operand or specified more than one DSNAME mask.

System action: Terminates with a condition code of 12.

Operator response: Correct the SYSIN file contents and rerun the program. To process multiple data set name masks via the CATALOG specify a separate

Inquisitor request for each mask. There is no programmed limit to the number of requests which can be processed in a single Inquisitor run.

System programmer response: None.

Module: HSIPINQ

HSIP063S ALL POSSIBLE STORAGE GROUPS ARE EXCLUDED

Explanation: An exclusion mask has been specified which excludes all possible storage groups included by the selection mask. Both masks are shown after this message.

System action: Terminates with a condition code of 12.

Operator response: Modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP064E ABEND *abend* **FOR** *mbr* **IN** *dsn*

Explanation: A subtask processing a program object from a PDSE has abended. The abend probably occurred in the Program Binder API.

In the message text:

abend
hexadecimal system abend code.

mbr
name of the member being processed.

dsn
name of the data set being processed.

System action: Data collected for this member so far is retained. Other Data Management abends may follow, especially in CLOSE processing, which are unrecoverable and may abend the main Inquisitor task.

Operator response: Exclude the programs causing the failure and rerun the Job.

System programmer response: None.

Module: HSIPINQ

HSIP065S MCDS FILE FAILED VERIFICAION

Explanation: The MCDS data definition (DD) was found to be unusable by the Inquisitor. One or more of the following is true: 1) The MCDS file could not be opened. Message HSIP066E follows. 2) The MCDS file is not a VSAM key-sequenced data set (KSDS). 3) The KSDS relative key position (RKP) is not zero (0). 4) The KSDS key length is not forty-four (44).

System action: Terminates with a condition code of 12.

Operator response: Either ensure that the Inquisitor has read access to DFHSM's MCDS, or change the Inquisitor request(s) so that the MCDS is not required. MCDS access is required if either or both of the REMIGRATE and NOML2 keywords are specified.

System programmer response: None.

Module: HSIPINQ

HSIP066E MCDS OPEN ERROR - RC=*rc* **RS=***rs*

Explanation: The OPEN of the MCDS data definition (DD) was not successful.

In the message text:

rc VSAM OPEN hexadecimal return code.

rs VSAM OPEN hexadecimal reason code.

System action: Issues message HSIP065S and terminates with a condition code of 12.

Operator response: Either ensure that the Inquisitor has read access to DFHSM's MCDS, or modify the Inquisitor request(s) so that the MCDS is not required. MCDS access is required if either or both of the REMIGRATE and NOML2 keywords are specified.

System programmer response: None.

Module: HSIPINQ

HSIP067E MCDS READ RC=*rc* **RS=***rs* **FOR** *dsn*

Explanation: The MCDS record of a data set cataloged on volume MIGRAT could not be read. Either the record is missing or there was an I/O error.

In the message text:

rc VSAM GET hexadecimal return code.

rs VSAM GET hexadecimal reason code.

dsn
name of data set cataloged on volume MIGRAT.

System action: Processing of this data set is terminated.

Operator response: If the data set is not really migrated then correct the catalog entry. If the MCDS is corrupt then begin recovery procedures.

System programmer response: None.

Module: HSIPINQ

HSIP068E CATALOG RC=*rc* **RS=***rs*,*modid cat*

Explanation: The Catalog Search Interface returned an entry which is flagged as being in error by Catalog Management.

In the message text:

rc Catalog Management decimal return code.

rs Catalog Management decimal reason code.

modid
Catalog Management module identifier.

cat
name of catalog entry in error.

System action: Processing of this data set is terminated.

Operator response: Correct the catalog entry. Refer to the System Messages manual for message IDC3009I to find out the meaning of the Catalog Management error codes.

System programmer response: None.

Module: HSIPINQ

HSIP069U PROGRAM IS NOT APF AUTHORIZED

Explanation: The Inquisitor has determined that it is not running in an APF authorized environment, and PARM=NOAPF was not specified.

System action: Terminates with a condition code of 20.

Operator response: Ensure that the HSIPINQ program is run in an APF authorized environment, or specify PARM=NOAPF in the JCL.

System programmer response: None.

Module: HSIPINQ

HSIP070E BAD BLKSIZE AFTER OPEN FOR *dsn*

Explanation: A BPAM DCB was opened for the named PDS, but despite the VTOC entry indicating a suitable blocksize, the blocksize in the DCB after the OPEN was not positive.

In the message text:

dsn
name of the data set being processed.

System action: Processing of member contents for this data set is terminated to avoid an S002-30 abend.

Operator response: The PDS is probably corrupt and should be deleted. Recreate it from a backup if appropriate.

System programmer response: None.

Module: HSIPINQ

HSIP071W IGNORING INVALID DSNAME IN *dsn*

Explanation: The Catalog Search Interface (CSI) returned a data set name with invalid characters. Although VTOC entries can contain keys that are not valid data set names, such entries cannot be cataloged. Therefore the entry returned from the CSI does not represent an actual data set.

In the message text:

dsn
name of the catalog being processed.

System action: The returned catalog entry is discarded.

Operator response: Ensure that the named catalog is not corrupt and contains no invalid entries.

System programmer response: None.

Module: HSIPINQ

HSIP072I BYPASS PROCESSING DATA SET *dsn*

Explanation: The name of the data set indicated that it does not contain programs which would normally be executed, and therefore the Inquisitor skipped processing it. This message is only issued when a program parameter of "DSNMSG" or "ALLMSG" is specified.

In the message text:

dsn
name of the data set being bypassed.

System action: The data set is not opened, and no data from it is collected.

Operator response: None required, but if the data set must be processed then specify its name in an inclusion mask without any generic masking characters, either by adding this mask to the existing request, or by adding an additional request to the same Inquisitor run.

System programmer response: None.

Module: HSIPINQ

HSIP073I NO DATA WAS EXTRACTED FROM *dsn*

Explanation: The data set contained no members eligible for selection. This message is only issued when a program parameter of "DSNMSG" or "ALLMSG" is specified.

In the message text:

dsn
name of the processed data set.

System action: The data set was opened, but no data from it is collected.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP074S ABRIN OR ABRPRINT FILES NOT ALLOCATED

Explanation: A request had ABRMIG and/or ABRARC specified but at least one of the required ABRIN and ABRPRINT files was not defined in the JCL.

System action: Terminates with a condition code of 12.

Operator response: Ensure the required files are pre-allocated for the Inquisitor.

System programmer response: None.

Module: HSIPINQ

HSIP075W FDRABR ABEND *abend* CHECKING *dsn*

Explanation: An abend occurred during ABR processing while checking a data set which may have been archived.

In the message text:

abend
hexadecimal system abend code.

dsn
name of the data set being processed.

System action: Processing of this data set is terminated.

Operator response: Ensure the catalog entry for the data set is correct.

System programmer response: None.

Module: HSIPINQ

HSIP076E BAD LOAD *abend*-*rs*: *mbr dsn*

Explanation: The Inquisitor attempted to load a product tag data module from the named data set, but LOAD issued the displayed abend code.

In the message text:

abend
abend code returned by LOAD.

rs abend reason code returned by LOAD.

mbr
name of the member containing the tag data.

dsn
name of the data set containing the tag data module.

System action: Processing continues with the next member in the data set.

Operator response: Verify that the named data set contains no unusable modules. If necessary, delete any modules that are of no further use.

System programmer response: None.

Module: HSIPINQ

HSIP077E ISITMGD RC=*rc* RS=*rs* FOR *dsn*

Explanation: The Inquisitor executed an ISITMGD macro for the named data set, but ISITMGD issued a non-zero return code.

In the message text:

rc decimal return code issued by ISITMGD.

rs hexadecimal reason code issued by ISITMGD.

dsn
name of the data set being processed.

System action: Processing continues with the next data set.

Operator response: Consult the applicable DFSMS Macro Instructions for Data Sets manual to determine the meaning of the ISITMGD return and reason codes. Ensure that the named data set is a valid and accessible partitioned data set. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP078E DESERV RC=*rc* RS=*rs* FOR *dsn*

Explanation: The Inquisitor executed a DESERV FUNC=GET_ALL macro for the named data set, but DESERV issued a non-zero return code.

In the message text:

rc decimal return code issued by DESERV.

rs hexadecimal reason code issued by DESERV.

dsn
name of the data set being processed.

System action: Processing continues with the next data set.

Operator response: Consult the applicable DFSMS Macro Instructions for Data Sets manual to determine the meaning of the DESERV return and reason codes. Ensure that the named data set is a valid and accessible partitioned data set. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIPINQ

HSIP080I DYNALLOC FAILURE: DSN=*dsn*

Explanation: A data set could not be dynamically allocated.

In the message text:

dsn
name of the data set being processed.

System action: Depends upon other messages associated with this message.

Operator response: None required.

System programmer response: None.

Module: HSIPINQ

HSIP081S ALL POSSIBLE DEVICE NUMBERS ARE EXCLUDED

Explanation: An exclusion mask has been specified which excludes all possible device numbers included by a selection mask. Both masks are shown after this message.

System action: Terminates with a condition code of 12.

Operator response: Modify or remove the conflicting selection criteria.

System programmer response: None.

Module: HSIPINQ

HSIP082I THE "XDEVICE" MASK IS NOT A SUBSET OF ANY "DEVICE" MASK

Explanation: The mask specified in the XDEVICE operand excludes possible values not included in the DEVICE operand. This message is issued to highlight possible inconsistencies in a request.

System action: Processing continues.

Operator response: Specify the XDEVICE operand as a further qualification of the DEVICE operand to avoid this message.

System programmer response: None.

Module: HSIPINQ

HSIP083E RENAME FAILED FOR DATA SET *dsn*

Explanation: The rename operation to add one or more extra low-level qualifiers to a data set name as specified by the LLQ program parameter setting did not succeed. The named data set is allocated to either the HSIPZIP or HSIPOUT file. This message is preceded by either an associated explanatory message, or by messages from IDCAMS detailing the results of the rename attempt.

In the message text:

dsn
name of the HSIPZIP or HSIPOUT data set.

System action: The output data set is left with its original name.

Operator response: Ensure that the specified LLQ string length does not exceed 44 bytes, that any symbols used are valid for this system, and that resultant data set names are not longer than 44 bytes. Examine associated messages to determine the reason for the rename failure.

System programmer response: None.

Module: HSIPINQ

HSIP097E CATALOG SEARCH INTERFACE ERROR RC=*csirc*

Explanation: A request with the CATALOG keyword was specified, and the Catalog Search Interface encountered an error.

In the message text:

csirc
return code from the Catalog Search Interface.

System action: Processing catalog entries for the request is terminated.

Operator response: Correct any related catalog errors.

System programmer response: None.

Module: HSIPINQ

HSIP098E CATALOG SEARCH INTERFACE ERROR RC=*csirc* **CATALOG RC=***rc* **CATALOG RS=***rs*

Explanation: A request with the CATALOG keyword was specified, and the Catalog Search Interface encountered an error.

In the message text:

csirc
return code from the Catalog Search Interface.

rc return code from Catalog Management.

rs reason code from Catalog Management.

System action: Processing catalog entries for the request is terminated.

Operator response: Correct any related catalog errors.

System programmer response: None.

Module: HSIPINQ

HSIP099E CATALOG SEARCH INTERFACE ERROR RC=*csirc* **CATALOG RC=***rc* **CATALOG RS=***rs* **MODULE=***modid*

Explanation: A request with the CATALOG keyword was specified, and the Catalog Search Interface encountered an error.

In the message text:

csirc
return code from the Catalog Search Interface.

rc return code from Catalog Management.

rs reason code from Catalog Management.

modid
module identifier.

System action: Processing catalog entries for the request is terminated.

Operator response: Correct any related catalog errors.

System programmer response: None.

Module: HSIPINQ

HSIP999U MODULE HSIPMSG FAILED - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Contact IBM Support.

Module: HSIPINQ :emsgl.

HSIT - Product Tagging messages

Return codes

0	No errors encountered. All requests processed successfully.
4	Warning issued. Processing continues. I/O error in one or more program libraries.
8	Error - Incomplete data. Processing continues. OPEN or system service error.
12	Severe error. Processing terminates. Utility failure or syntax error.
16	Unrecoverable error. No requests processed. SYSIN file cannot be used.
20	Disastrous error. No requests processed. SYSPRINT file cannot be used.

Message suffix codes

Suffix	Meaning	Raises Minimum Condition Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIT001U HSITAGP COULD NOT OPEN THE INPUT FILE *file*

Explanation: A required file could not be opened successfully.

In the message text:

file
name of file.

System action: Processing terminates with condition code 16.

Operator response: Correct the file definition and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT002S UNRECOGNIZED STATEMENT TYPE: *stattyp*

Explanation: Input text was encountered which does not match any known statement type.

In the message text:

stattyp
encountered input data.

System action: Processing terminates with condition code 12.

Operator response: Correct the input and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT003S DUPLICATE VALUE SUPPLIED FOR *stattyp*

Explanation: More than one occurrence of the named statement type was encountered, but only one value can be accepted.

In the message text:

stattyp
name of the statement verb.

System action: Processing terminates with condition code 12.

Operator response: Remove the redundant statement and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT004S VALUE MISSING IN *stattyp* **STATEMENT**

Explanation: An input statement of the type indicated was encountered, but no non-blanks followed the statement type name.

In the message text:

stattyp
name of the statement verb.

System action: Processing terminates with condition code 12.

Operator response: Supply an appropriate value after the statement type name.

System programmer response: None.

Module: HSITAGP

HSIT005S VALUE SPECIFIED FOR LICENSED WAS NEITHER "YES" NOR "NO"

Explanation: A LICENSED statement was processed which had a value specified other than one of the valid values.

System action: Processing terminates with condition code 12.

Operator response: Correct the value and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT006S THE *parm* **PARAMETER HAD NO SUBPARAMETER VALUE SPECIFIED**

Explanation: A statement parameter or operand was specified, but the required subparameter, or value of the parameter, was not specified. One cause for this condition is the omission of a parenthesis.

In the message text:

parm
name of the parameter or operand being processed when the error is detected.

System action: Processing terminates with condition code 12.

Operator response: Correct the input and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT007I A CLOSING PARENTHESIS ASSUMED FOR *parm*

Explanation: End-of-file was raised when processing input statements before an expected close parenthesis was encountered.

In the message text:

parm
name of the parameter or operand being processed when the error is detected.

System action: Processing continues as if the expected close parenthesis had been specified.

Operator response: Check that the resulting processing is as expected. Correct the input file for future use, and rerun the job if the desired processing was not performed.

System programmer response: None.

Module: HSITAGP

HSIT008S UNEXPECTED OPEN PARENTHESIS ENCOUNTERED AFTER *parm*

Explanation: An open parenthesis was encountered when one was not expected. If this occurred while a parameter or operand was being processed, then it is named in the message.

In the message text:

parm
name of the parameter or operand being processed when the error is detected.

System action: Processing terminates with condition code 12.

Operator response: Correct the input file and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT009S UNEXPECTED CLOSE PARENTHESIS ENCOUNTERED AFTER *parm*

Explanation: A close parenthesis was encountered when one was not expected. If this occurred while a parameter or operand was being processed, then it is named in the message.

In the message text:

parm
name of the parameter or operand being processed when the error is detected.

System action: Processing terminates with condition code 12.

Operator response: Correct the input file and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT010S *parm* **IS AN UNKNOWN TAGLIBS PARAMETER**

Explanation: Input data was encountered which is not a recognized parameter, or operand, of the TAGLIBS statement.

In the message text:

parm
the encountered input data.

System action: Processing terminates with condition code 12.

Operator response: Correct the input file and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT011S MEMBER NAME *parm* **HAS EMBEDDED BLANK(S)**

Explanation: The value specified on the TAGMEM statement was not a valid partitioned data set member name, a blank was found within the eight character member name.

In the message text:

parm
the input value specified on the TAGMEM statement.

System action: Processing terminates with condition code 12.

Operator response: Correct the input file and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT012S MISSING OPEN PARENTHESIS AFTER *parm*

Explanation: Whilst parsing the TAGLIBS statement looking for a subparameter, or value, in parentheses specified for the parameter or operand named in the message, text was encountered which was not enclosed in parentheses.

In the message text:

parm
name of the parameter or operand being processed when the error is detected.

System action: Processing terminates with condition code 12.

Operator response: Correct the input and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT013S VALUE *data* **TOO LONG FOR PARAMETER** *parm*

Explanation: The length of a subparameter or value was found to exceed the maximum length allowed. The maximum length allowed depends on the specific parameter or operand being processed. For example, a data set name mask exceeding 44 characters in length causes this condition, as will a volume mask exceeding six characters in length.

In the message text:

data
encountered input data.

parm
name of the parameter or operand being processed when the error is detected.

System action: Processing terminates with condition code 12.

Operator response: Correct the input and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT014S END OF INPUT REACHED, EXPECTED CONTINUATION IS MISSING

Explanation: End-of-file was raised on the input (SYSIN) file, but the TAGLIBS statement currently being processed was expected to continue on the next record.

System action: Processing terminates with condition code 12.

Operator response: Either supply the missing input data, or remove the continuation character from the last input record. Rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT015S REQUIRED DATA SET NAME SPECIFICATION IS MISSING

Explanation: The processing of a TAGLIBS statement completed without encountering a data set name selection mask specification.

System action: Processing terminates with condition code 12.

Operator response: Ensure that all TAGLIBS statements specify at least one value in a DATASET, DSNAME parameter, or operand.

System programmer response: None.

Module: HSITAGP

HSIT016S SYMBOL SUBSTITUTION FAILURE - ASASYMBM RC=*rc*

Explanation: Symbol substitution was attempted for a TAGLIBS statement record which had at least one ampersand in it, and the system symbol substitution routine ASASYMBM terminated with a non-zero return code.

In the message text:

rc the completion code returned by ASASYMBM.

System action: Processing terminates with condition code 12.

Operator response: Check all uses of symbols in the input (SYSIN) file. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSITAGP

HSIT017S NO VALUE FOR *stattyp* WAS SPECIFIED

Explanation: A value for a statement of the type named in the message is required, but was not found in the input file.

In the message text:

stattyp
the type of input statement required to specify the missing value.

System action: Processing terminates with condition code 12.

Operator response: Supply a statement of the named type which specifies a value.

System programmer response: None.

Module: HSITAGP

HSIT018W CATALOG SEARCH TERMINATED BY RC=*rc* WHILE SEARCHING FOR *mask*

Explanation: The Catalog Search Interface (CSI) was called to search for catalog entries matching the displayed data set name mask, but the CSI call ended with a non-zero return code. This may indicate a corrupted catalog or trouble accessing a potentially relevant catalog. It means that one or more data sets which you intended to process were not processed.

In the message text:

rc the return code issued by the Catalog Search Interface.

mask
the data set name mask passed to the Catalog Search Interface.

System action: Processing continues with any data set catalog entries that can be accessed.

Operator response: Check the data set name mask for suitability to your system environment. Use IDCAMS to examine any relevant catalog entries. Ensure that all catalogs which would be referenced by such a search are on accessible volumes. Use the same data set name mask with other software which also calls the CSI, such as the Inquisitor, or ISPF option 3.4, to see if further diagnostic information can be acquired. The error will usually be associated with the local catalog configuration and catalog contents.

System programmer response: None.

Module: HSITAGP

HSIT019E DESERV FAILED - RC=*rc* RS=*rs* FOR DATA SET *dsn*

Explanation: A DESERV FUNC=GET_ALL macro was issued to acquire the member list for a data set, but DESERV issued a non-zero return code.

In the message text:

rc the decimal return code issued by DESERV.

rs the hexadecimal reason code issued by DESERV.

dsn
the name of the data set being processed by DESERV.

System action: The named data set is not processed, and processing continues with the next relevant data set.

Operator response: Consult the applicable DFSMS Macro Instructions for Data Sets manual to determine the meaning of the DESERV return and reason codes. Ensure that the named data set is a valid and accessible program library. If necessary, gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSITAGP

HSIT020S DYNAMIC ALLOCATION FAILURE - BPXWDYN RC=*rc*

Explanation: BPXWDYN was called to dynamically allocate a required work file, but BPXWDYN issued a non-zero return code. As a result, processing cannot proceed.

In the message text:

rc the hexadecimal return code issued by BPXWDYN.

System action: Processing terminates with condition code 12.

Operator response: Consult the applicable Using REXX and z/OS UNIX System Services manual to determine the meaning of the return code. Examine the job log and messages to see any associated dynamic allocation error message.

System programmer response: None.

Module: HSITAGP

HSIT021E DYNALLOC FAILED - RC=*rc* **ERROR=***s99err* **INFO=***s99inf* **FOR DATA SET** *dsn*

Explanation: A DYNALLOC macro was issued to dynamically allocate a program library for processing, but DYNALLOC issued a non-zero return code.

In the message text:

rc the decimal return code issued by DYNALLOC.

s99err
the contents of S99ERROR in hexadecimal.

s99inf
the contents of S99INFO in hexadecimal.

dsn
name of the data set which could not be allocated.

System action: Processing continues with the next data set.

Operator response: Consult the applicable MVS Authorized Assembler Services Guide to determine the meaning of the dynamic allocation return code, error, and information codes. Check that the named data set is accessible and available for allocation.

System programmer response: None.

Module: HSITAGP

HSIT022S RC=*rc* **WAS RETURNED BY PROGRAM** *pgm*

Explanation: Either the High Level Assembler (program ASMA90) or the Program Binder (program IEWL) was dynamically started to assist with creating the output data, but the named program issued a non-zero return code.

In the message text:

rc the decimal return code issued by the named program.

pgm
the name of the program that was started.

System action: Processing terminates with condition code 12.

Operator response: Examine all associated job output to determine if the problem is caused by a correctable environmental error. If so, make the correction and rerun the job. If not, gather all relevant diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSITAGP

HSIT023I PROCESSING TERMINATED DUE TO ENCOUNTERED ERROR CONDITION

Explanation: Because of a previously reported error, the Product Tagging Utility is terminating unilaterally, without processing all of the specified program library data sets, and without generating all of the requested program product tagging data.

System action: Processing terminates.

Operator response: Investigate any previously reported error conditions.

System programmer response: None.

Module: HSITAGP

HSIT024E ISITMGD FAILED - RC=*rc* **RS=***rs* **FOR FILE** *file* **AND DATA SET** *dsn*

Explanation: An ISITMGD macro was issued against a program library, but ISITMGD issued a non-zero return code.

In the message text:

rc the decimal return code issued by ISITMGD.

rs the decimal reason code issued by ISITMGD.

file
the name of the file being processed by ISITMGD.

dsn
the name of the data set being processed by ISITMGD.

System action: The named data set is not processed, and processing continues with the next relevant data set.

Operator response: Consult the applicable DFSMS Macro Instructions for Data Sets manual to determine the meaning of the ISITMGD return and reason codes. Ensure that the named data set is a valid and accessible partitioned data set. If necessary, gather the appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSITAGP

HSIT025I *pgmcnt* **PROGRAMS FOUND TO TAG FROM DATA SET** *dsn*

Explanation: Input processing of the named data set has completed, resulting in data from the reported number of programs being accumulated for subsequent output.

In the message text:

pgmcnt
the number of programs processed.

dsn
the data set name containing the processed programs.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSITAGP

HSIT026I **PROCESSING COMPLETE - RC=***rc* **AND** *pgmcnt* **PROGRAMS TAGGED IN TOTAL**

Explanation: The Product Tagging Utility program HSITAGP has completed processing. This message reports the return code issued by HSITAGP, and the number of programs from which data has been collected during this run.

In the message text:

rc the return code issued by the HSITAGP upon termination.

pgmcnt
the number of programs processed in this run of HSITAGP.

System action: Processing is completed with the displayed return code.

Operator response: None required.

System programmer response: None.

Module: HSITAGP

HSIT027E **DYNALLOC FAILED - RC=***rc* **ERROR=***s99err* **INFO=***s99inf* **FOR VOLUME** *vol*

Explanation: A DYNALLOC macro was issued to dynamically allocate a volume for processing, but DYNALLOC issued a non-zero return code.

In the message text:

rc the decimal return code issued by DYNALLOC.

s99err
the contents of S99ERROR in hexadecimal.

s99inf
the contents of S99INFO in hexadecimal.

vol
the volume serial number of the volume which could not be allocated.

System action: Processing continues with the next volume.

Operator response: Consult the applicable MVS Authorized Assembler Services Guide to determine the meaning of the dynamic allocation return code, error, and information codes. Check that the named volume is accessible and available for allocation.

System programmer response: None.

Module: HSITAGP

HSIT028W **UNABLE TO ACQUIRE ANY PRODUCT MAINTENANCE LEVEL DATE**

Explanation: After having processed all of the relevant programs, HSITAGP was unable to acquire any date stamp for use as a maintenance level indicator.

System action: Blanks are placed in the maintenance level field and processing continues.

Operator response: None required.

System programmer response: None.

Module: HSITAGP

HSIT029S *stattyp* **STATEMENT VALUE LENGTH EXCEEDS THE ALLOWED MAXIMUM OF** *max* **BYTES**

Explanation: The value specified for the named statement type was found to be longer than the maximum allowed. The maximum byte count allowed

for a value of this statement type is shown in the message.

In the message text:

stattyp
the type of input statement being processed.

max
number of bytes.

System action: Processing terminates.

Operator response: Correct the input and rerun the job.

System programmer response: None.

Module: HSITAGP

HSIT030S INVALID TEXT CHARACTER X"*char*" FOUND IN *stattyp* STATEMENT

Explanation: The displayed data byte was encountered when processing the value specified for the statement type indicated. The value specified on the statement is expected to be a string. Valid byte values for text data are in the range from X'40' to X'FE' inclusive. The control code encountered is either not valid input, or not valid input in this location. The only control codes that can be used in the input value are SO (X'0E') and SI(X'0F'), when they are used to encapsulate DBCS data.

In the message text:

char
the hexadecimal value of the invalid text code point.

stattyp
the type of input statement being processed.

System action: Processing terminates.

Operator response: Remove the undisplayable characters from the input value. If using DBCS, ensure that SO precedes DBCS text and SI terminates DBCS text, and that the DBCS text is an even number of valid text bytes.

System programmer response: None.

Module: HSITAGP

HSIT031S SCANNING FOUND MULTIPLE VERSIONS OF PROGRAM *pgm*

Explanation: A duplicate named program with different characteristics was detected in the tag scan. Consequently, no tag data members were output to prevent subsequent IQ Import errors.

In the message text:

pgm
name of program.

System action: Processing terminates with condition code 8.

Operator response: Delete unneeded copies of the program. If necessary, exclude the named program from the tag scan.

System programmer response: None.

Module: HSITAGP

HSIT999U HSIMSG/HSITMSG FAILURE - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Ensure Joblib/Steplib contains the library where the HSITMSG message module resides. If you cannot resolve this issue then contact IBM support.

Module: HSITAGP :emsgl.

HSIX - Inquisitor for z/OS UNIX messages and codes

Return codes

0	No errors encountered. All requests processed successfully.
4	I/O error in one or more program libraries.
8	Error - Incomplete data. Processing continues. OPEN or other system service error.
12	Syntax error.
16	Unrecoverable error. No requests processed. SYSIN or HSIPZIP or HSIPOUT File cannot be used, or unsupported Operating System.

20	Disastrous error. No requests processed. SYSPRINT file cannot be used.

Message suffix codes

Suffix	Meaning	Raises Minimum Condition Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIX002I THE SPECIFIED DIRECTORY NAME DOES NOT START WITH A SLASH

Explanation: A record from file HSIXROOT was read and was found to start with a non-blank that is not a slash. It is flagged in case processing errors result from the non-standard directory name.

System action: Processing continues.

Operator response: Correct the input if it is incorrect.

System programmer response: None.

Module: HSIXINQ

HSIX003I PROGRAM PARAMETER "*parm*" DISCARDED

Explanation: The program parameter contained some unrecognized data.

In the message text:

parm
parameter in error.

System action: The displayed part of the program parameter is ignored.

Operator response: Correct the program parameter.

System programmer response: None.

Module: HSIXINQ

HSIX004I FUNCTION *func* FAILED, HEX RC=*rc*, HEX REASON=*rs*

Explanation: The named z/OS UNIX system service issued a negative return code.

In the message text:

func
function name.

rc return Code.

rs reason Code.

System action: Processing continues.

Operator response: Determine the meaning of the return and reason code values, and correct the problem if appropriate. Information relating to the failing UNIX function can be found in the UNIX System Services Assembler Callable Services manual. Information relating to the Return Code and Reason Code of the failing UNIX function can be found in the UNIX System Services Messages and Codes manual.

System programmer response: None.

Module: HSIXINQ

HSIX005E *pth*

Explanation: The named path was not successfully processed by the z/OS UNIX system service named in the preceding HSIX004I message. Data will not be collected from directories and files which could not be opened.

In the message text:

pth
path in error.

System action: Processing continues.

Operator response: Ensure that all necessary parts of the z/OS UNIX file system are accessible.

System programmer response: None.

Module: HSIXINQ

HSIX006E RENAME FAILED FOR DATA SET *dsn*

Explanation: The rename operation to add one or more extra low-level qualifiers to a data set name as specified by the LLQ program parameter setting did

not succeed. The named data set is allocated to either the HSIXZIP or HSIXOUT file. If this message is not followed by an associated explanatory message then an IDCAMS report detailing the results of the rename attempt will have been written to SYSPRINT.

In the message text:

dsn
name of the HSIXZIP or HSIXOUT data set.

System action: The output data set is left with its original name.

Operator response: Ensure that the specified LLQ string length does not exceed 44 bytes, that any symbols used are valid for this system, and that resultant data set names are not longer than 44 bytes. Examine associated messages to determine the reason for the rename failure.

System programmer response: None.

Module: HSIXINQ

HSIX999U HSIMSG/HSIXMSG FAILURE - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Ensure Joblib/Steplib contains the library where the HSIXMSG message module resides. If you cannot resolve this issue then contact IBM support.

Module: HSIXINQ :emsgl.

HSIZ - Usage Monitor messages

Return codes

0	Normal termination.
16	Initialization failed.

Message suffix codes

Suffix	Meaning	Raises Minimum Cond Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIZ001I USAGE MONITOR INITIALIZING

Explanation: The Usage Monitor has been started.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ002I *csid* DETECTED UNSUPPORTED OPERATING SYSTEM

Explanation: The Usage Monitor may not run on an unsupported operating system.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ003I *csid* **USAGE MONITOR NOT APF AUTHORIZED**

Explanation: The Usage Monitor needs to be executed in an APF authorized environment.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: See System Programmer to correct the error.

System programmer response: APF authorize the load libraries that the Usage monitor runs from.

Module: HSIZMON

HSIZ005I *csid* **USAGE MONITOR ALREADY ACTIVE**

Explanation: The Usage Monitor is already running. Only one concurrent copy can run in an operating system image.

In the message text:

csid
current system identifier.

System action: Processing terminates. The established Usage Monitor task continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ006I *csid* **USAGE MONITOR QEDIT BUFFER SET FAILED**

Explanation: A QEDIT issued to set up MODIFY command processing has failed.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: Notify the system programmer.

System programmer response: Gather appropriate diagnostic materials and contact IBM support.

Module: HSIZMON

HSIZ007I *csid* **USAGE MONITOR MODULE** *mod* **FAILED - RC=***rc*

Explanation: A Usage Monitor subroutine has failed.

In the message text:

csid
current system identifier.

mod
failing module name.

rc decimal return code.

System action: Processing terminates.

Operator response: Notify the system programmer.

System programmer response: If the return code is 312, then you must increase your MAXCAD parameter. This requires an IPL. For any other return codes, gather appropriate diagnostic materials and contact IBM support.

Module: HSIZMON

HSIZ008I *csid* **USAGE MONITOR ASID** *asid* **SET IN AVT** *avt*

Explanation: An Anchor Vector Table (AVT) has been acquired or reacquired, and has been updated for the current server address space.

In the message text:

csid
current system identifier.

asid
ASID number.

avt
AVT Address.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ009I **DATA WRITTEN TO DSN=***dsn*

Explanation: Usage Monitor data has been written to the named data set.

In the message text:

dsn
data set name of the created output.

System action: Processing continues.

Operator response: Transfer the named data set to the system where the database resides so it can be processed.

System programmer response: None.

Module: HSIZMON

HSIZ010E *csid* **USAGE MONITOR - WRITER TASK ENDED - RC=***rc*

Explanation: A writer task has ended with a non-zero return code.

In the message text:

csid
current system identifier.

rc return code of writer task.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Gather appropriate diagnostic materials and contact IBM support.

Module: HSIZMON

HSIZ011E *csid* **USAGE MONITOR - WRITER TASK ABENDED - S***abend*

Explanation: A writer task has ended abnormally.

In the message text:

csid
current system identifier.

abend
abend code from writer task.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Local reasons for system abends should be investigated. If necessary, gather appropriate diagnostic materials and contact IBM support.

Module: HSIZMON

HSIZ012I ****DATA LOSS** UNUSABLE DSN=***dsn*

Explanation: It is likely that Usage Monitor data has been lost because of unexpected behaviour by a writer task. Any compressed output data that has been written will probably be unusable.

In the message text:

dsn
data set name of the created output file.

System action: Processing continues.

Operator response: Examine any preceding messages to determine the likely cause of the writer task error. If the output data set is complete it can be used, otherwise if the data is compressed it is unusable. If the data set is empty then this fact can be noted and the data set can be deleted. Unless retaining an unusable data set for diagnosis reasons it can be deleted.

System programmer response: Investigate any writer task abends.

Module: HSIZMON

HSIZ013I *csid* **USAGE MONITOR - UNRECOGNISED PROGRAM PARAMETER IGNORED**

Explanation: An unrecognised program parameter was specified.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: Remove or correct the program parameter.

System programmer response: None.

Module: HSIZMON

HSIZ014I *csid* **USAGE MONITOR - COULD NOT OPEN FILE HSIZIN**

Explanation: The HSIZIN file could not be opened by the Usage Monitor.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: Supply or correct the HSIZIN DD statement in the JCL.

System programmer response: None.

Module: HSIZMON

HSIZ015I *csid* **USAGE MONITOR - COULD NOT OPEN FILE HSIZMSG**

Explanation: The HSIZMSG file could not be opened by the Usage Monitor.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: Supply or correct the HSIZMSG DD statement in the JCL.

System programmer response: None.

Module: HSIZMON

HSIZ016I *csid* **USAGE MONITOR TERMINATING - INVALID OR MISSING HSIZIN DATA**

Explanation: At least one HSIZIN input statement was invalid, or input required to be present in the HSIZIN file was missing.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: Examine the HSIZMSG output report. Correct any invalid statements. Ensure a valid data set name prefix was specified.

System programmer response: None.

Module: HSIZMON

HSIZ017I *csid* **USAGE MONITOR TERMINATING - NOW WRITING CAPTURED DATA**

Explanation: A STOP command has been encountered. The current repository contents are written before the Usage Monitor terminates.

In the message text:

csid
current system identifier.

System action: The Usage Monitor starts a writer task and waits for its completion before terminating.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ018I *csid* **USAGE MONITOR HAS NOW TERMINATED**

Explanation: The Usage Monitor has now freed resources and is about to terminate.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ019I *csid* **USAGE MONITOR REPOSITORY FULL - NOW SWITCHING**

Explanation: The current Usage Monitor data collection repository is full.

In the message text:

csid
current system identifier.

System action: A new repository is created and used for subsequent data collection. A writer task is initiated for the full repository.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ020I *csid* **THE SPECIFIED NUMBER WAS TOO SMALL**

Explanation: The numeric value of a command subparameter was too small to be valid in the command context.

In the message text:

csid
current system identifier.

System action: The command is discarded.

Operator response: Correct the numeric value and reissue the command.

System programmer response: None.

Module: HSIZMON

HSIZ021I *csid* **THE SPECIFIED NUMBER WAS TOO LARGE**

Explanation: The numeric value of a command subparameter was too large to be valid in the command context.

In the message text:

csid
current system identifier.

System action: The command is discarded.

Operator response: Correct the numeric value and reissue the command.

System programmer response: None.

Module: HSIZMON

HSIZ022I *csid* **PASSIVE MODE SET FROM PROGRAM PARAMETER**

Explanation: PASSIVE was specified in the program parameter.

In the message text:

csid
current system identifier.

System action: The Usage Monitor starts in passive mode unless overridden by input from the HSIZIN file.

Operator response: Set the Usage Monitor into collection mode to start data collection.

System programmer response: None.

Module: HSIZMON

HSIZ023I *csid* **PROGRAM NAME MASK** *mask* **NOT ADDED - ALREADY IN TABLE**

Explanation: A command to add a program name mask to a program mask table was issued, but the mask was already present in the table.

In the message text:

csid
current system identifier.

mask
program mask specified in command.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ024I *csid* **PROGRAM NAME MASK** *mask* **ADDED TO TABLE**

Explanation: A command to add a program name mask to a program mask table was issued, and the mask was added successfully.

In the message text:

csid
current system identifier.

mask
program mask specified in command.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ025I *csid* **PROGRAM NAME MASK** *mask* **NOT DELETED - NOT FOUND IN TABLE**

Explanation: A command to delete a program name mask from a program mask table was issued, but the mask was not present in the table.

In the message text:

csid
current system identifier.

mask
program mask specified in command.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ026I *csid* **PROGRAM NAME MASK** *mask* **DELETED FROM TABLE**

Explanation: A command to delete a program name mask to a program mask table was issued, and the mask was deleted successfully.

In the message text:

csid
current system identifier.

mask
program mask specified in command.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ027I **ECSA APPEARS TO BE EXHAUSTED - INCREASE SIZE FOR NEXT IPL**

Explanation: The Usage Monitor has attempted to acquire storage from ECSA, but was given CSA storage by the system. This indicates that there is insufficient ECSA for the current workloads, and that it should be increased for the next IPL.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Add around 50 to 100 megabytes to the ECSA size in the system IPL parameters. Check the capacity of the COMMON page data set.

Module: HSIZMON

HSIZ028I **ECSA AND CSA APPEAR TO BE EXHAUSTED - INCREASE ECSA NEXT IPL**

Explanation: The Usage Monitor has attempted to acquire some common storage, but the requested amount was unavailable. This indicates that there is insufficient ECSA for the current workloads, and that it should be increased for the next IPL.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Add around 50 to 100 megabytes to the ECSA size in the system IPL parameters. Close some applications using CSA. If necessary, commence orderly shutdown and reIPL before the system crashes. Check the capacity of the COMMON page data set.

Module: HSIZMON

HSIZ029I *csid* **THERE IS CURRENTLY NO EXCLUDE TABLE**

Explanation: A request was made to change or display the program name mask exclude table, but there is currently no exclude table.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: None required. The EXC command may be used to create a table.

System programmer response: None.

Module: HSIZMON

HSIZ030I *csid* **USAGE MONITOR - NO DATA COLLECTED SO SKIPPING WRITE**

Explanation: Before a writer task was initiated to output the contents of a Usage Monitor repository, it was found that the repository contained no data, and that therefore data output processing could be omitted.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ031I *csid* **INITIATING REPOSITORY SWITCH**

Explanation: A switch (SWI) command was issued and the requested action is being initiated.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ032I *csid cmd* **COMMAND UNKNOWN**

Explanation: A command was issued but was not recognised.

In the message text:

csid
current system identifier.

cmd
name of the issued command.

System action: The command is ignored. Processing continues.

Operator response: If necessary, correct and reissue the command.

System programmer response: None.

Module: HSIZMON

HSIZ033I *csid cmd* **COMMAND PROCESSED**

Explanation: A command was issued and has been processed successfully.

In the message text:

csid
current system identifier.

cmd
name of the issued command.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ034I *csid cmd* **COMMAND HAS INVALID OPERAND**

Explanation: A command was issued but an invalid operand was encountered.

In the message text:

csid
current system identifier.

cmd
name of the issued command.

System action: The command is ignored. Processing continues.

Operator response: If necessary, correct and reissue the command.

System programmer response: None.

Module: HSIZMON

HSIZ035I *csid cmd* **COMMAND FAILED**

Explanation: A command was issued but insufficient resources were available to execute it successfully.

In the message text:

csid
current system identifier.

cmd
name of the issued command.

System action: The command is ignored. Processing continues.

Operator response: Try again after more resources become available.

System programmer response: None.

Module: HSIZMON

HSIZ036I *csid cmd* **COMMAND CAUSED NO CHANGE**

Explanation: A command was issued but the state to be set by the command was found to already exist.

In the message text:

csid
current system identifier.

cmd
name of the issued command.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ037I *csid cmd* **COMMAND REJECTED**

Explanation: A recognised command was issued at a time when the Usage Monitor is unable to process the command.

In the message text:

csid
current system identifier.

cmd
name of the issued command.

System action: The command is ignored. Processing continues.

Operator response: Try again after the Usage Monitor has freed the resources.

System programmer response: None.

Module: HSIZMON

HSIZ038I *csid* **CURRENT USAGE MONITOR PROGRAM EXCLUDE LIST:**

Explanation: A D-X command was issued to display the program name exclude table contents. The active entries are shown after this message.

In the message text:

csid
current system identifier.

System action: The data is displayed and processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ039I *csid* **REPOSITORY SWITCH HAS BEEN QUEUED**

Explanation: A repository switch was triggered by a SWI or STOP command, or by the current repository becoming full, but a writer task is already active. This message is followed by message HSIZ040I which shows the creation timestamp of the active writer task.

In the message text:

csid
current system identifier.

System action: Data collection is suspended. Wait for the current writer task to complete whereupon a new writer task is created, and a new repository is created, and data collection is resumed.

Operator response: Check that there are sufficient resources to dispatch the Usage Monitor address space. Check that there are no serialization problems with system components such as device allocation which could be inhibiting writer task processing.

System programmer response: None.

Module: HSIZMON

HSIZ040I *csid* **WAITING FOR WRITER TASK ATTACHED** *ts*

Explanation: A repository switch was triggered by a SWI or STOP command, or by the current repository becoming full, but a writer task is already active. This message follows message HSIZ039I and shows the creation timestamp of the active writer task.

In the message text:

csid
current system identifier.

ts Time stamp of write task.

System action: Data collection is suspended. Wait for the current writer task to complete whereupon a new

writer task is created, and a new repository is created, and data collection is resumed.

Operator response: Check that there are sufficient resources to dispatch the Usage Monitor address space. Check that there are no serialization problems with system components such as device allocation which could be inhibiting writer task processing.

System programmer response: None.

Module: HSIZMON

HSIZ041I *csid* **CURRENT USAGE MONITOR OUTPUT DYNALLOC PARMS:**

Explanation: A D-A command was issued to display the current output dynamic allocation parameters, which are shown after this message.

In the message text:

csid
current system identifier.

System action: The data is displayed and processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ042I **CURRENT USAGE MONITOR OUTPUT SYSTEM ID IS "***csid***"**

Explanation: A D-I command was issued to display the current system identifier which is to be contained in output header records.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ043I *csid* **DATA DISCARDED DUE TO (E)CSA STORAGE LIMIT**

Explanation: The Usage Monitor has detected for the first time in the life of the repository or since a CSA limit change that program usage event data has been discarded due to the CSA/ECSA storage usage limit being reached. This limit was set with the CSA command.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: Adjust the Usage Monitor CSA limit as appropriate for the particular system. Ensure that the ECSA size has been generously defined for the system, and that the common page data set size is adequate. Ensure that the Usage Monitor address space is running at a higher priority than all CPU-bound workloads. Generally, monitors need to run at a higher priority than the workloads being monitored.

System programmer response: None.

Module: HSIZMON

HSIZ044I *csid* **SWITCH-AND-WRITE TIME-OF-DAY IS SET TO** *hh***:***mm*

Explanation: A D-T command was issued to display the switch-and-write time-of-day setting for this system.

In the message text:

csid
current system identifier.

hh Hour of the day.

mm minute of the hour.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ045I *csid* **CREATED REPOSITORY** *token-alet*

Explanation: A repository was created to hold collected program usage data.

In the message text:

csid
current system identifier.

token
space token of the repository data space.

alet
ALET of the repository data space.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ046I *csid* **DELETED REPOSITORY** *token-alet* *nbr* **ENTRIES CACHED**

Explanation: A repository which was no longer needed was deleted.

In the message text:

csid
current system identifier.

token
space token of the repository data space.

alet
ALET of the repository data space.

nbr
number of entries cached to reduce overhead.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ047I *csid* **USAGE MONITOR - ATTACHING WRITER SEQ-NO-***seqnbr*

Explanation: A writer task is being attached to write out repository contents. The writer task sequence number is also reported. The first writer task to run after the Usage Monitor starts has a sequence number of 1.

In the message text:

csid
current system identifier.

seqnbr
sequence number of writer task this run.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ048I *csid* **USAGE MONITOR - IDENTIFY FAILED HEX RC=***rc*

Explanation: The Usage Monitor executed an IDENTIFY macro which failed.

In the message text:

csid
current system identifier.

rc hexadecimal return code of the IDENTIFY macro.

System action: Processing terminates.

Operator response: Notify the system programmer.

System programmer response: Investigate why an IDENTIFY macro would fail with that return code.

Module: HSIZMON

HSIZ049I *csid* **DATA SET NAME MASK NOT DEACTIVATED, NOT FOUND IN LIST**

Explanation: A command to delete a data set name mask from a data set name mask list was issued, but the mask was not present in the list.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ050I *csid* **DATA SET NAME MASK** *mask* **LIST** *list*

Explanation: A D-D command was issued to display the data set name mask include and exclude lists. These header and trailer lines mark the start and end of the lists.

In the message text:

csid
current system identifier.

mask
INCLUDE or EXCLUDE.

list
START or END.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ051I *csid* **(E)CSA QUEUING STORAGE LIMIT:** *limit*

Explanation: Either a CSA command was issued to change the limit setting, or a D-S command was issued. The CSA queuing storage limit can be used to limit the quantity of CSA to be used to hold program usage data elements queued for storing into the data space repository. When this limit is reached further data is discarded. A count of discarded elements is maintained and reported at termination. A limit of zero means the usage monitor never tries to limit CSA storage usage.

In the message text:

csid
current system identifier.

limit
no limit or kilobyte limit.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ052I *csid* **THE CACHE TRIGGER EVENT COUNT IS** *cnt*

Explanation: Either a TRG command was issued to change the cache trigger event count or a D-S command was issued. When a job uses the same program a number of times, the repository entry may be cached if the number has reached the cache trigger event count. The updating of cached entries is a synchronous process which does not use common storage. A limited number of cache entries are available. The cache is cleared when the repository is switched.

In the message text:

csid
current system identifier.

cnt
event count required to cause a program usage entry to be cached.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ053I *csid* **MONITORING UNIX PROGRAMS?** *ans*

Explanation: Either a USS command was issued to change the UNIX program monitoring status or a D-S command was issued. When the answer is YES the usage of programs fetched from UNIX files is monitored. When the answer is NO only the usage of programs from PDS and PDSE libraries is monitored.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ054I *csid* **MONITORING LINK PACK AREA PROGRAMS?** *ans*

Explanation: Either an LPA command was issued to change the LPA program monitoring status or a D-S command was issued. When the answer is YES the usage of programs residing in the Link Pack Area is monitored. When answer is NO only the usage of programs loaded into address space regions (and sometimes into CSA) is monitored.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ056I *csid* **PREFER VOLUME SYMBOL OVER SERIAL?** *ans*

Explanation: Either a SYM command was issued to change the volume symbol status or a D-S command was issued. When the answer is YES a matching system static symbol which evaluates to the volume serial is collected instead of the volume serial if such a symbol exists, otherwise the actual volume serial is collected. When the answer is NO the captured volume serial number is always output. A YES setting may be useful to improve data matching when system software platform volume switches take place.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ058I *csid* **FILE HSIZIN IS NOT ALLOCATED - CANNOT PERFORM REFRESH**

Explanation: A REF command was issued to refresh settings from commands in the HSIZIN file, but the HSIZIN file had been freed, and was no longer allocated to the Usage Monitor.

In the message text:

csid
current system identifier.

System action: The refresh operation is suppressed and processing continues.

Operator response: Ensure FREE=CLOSE is not specified in the HSIZIN JCL DD statement. Recycle the Usage Monitor to refresh the settings if necessary.

System programmer response: None.

Module: HSIZMON

HSIZ059I *csid* **REFRESH PERFORMED WITH NO ERRORS**

Explanation: A REF command was issued to refresh settings from commands in the HSIZIN file. All commands in the HSIZIN file were completed successfully.

In the message text:

csid
current system identifier.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ060I *csid* **REFRESH PERFORMED BUT ERROR(S) FOUND**

Explanation: A REF command was issued to refresh settings from commands in the HSIZIN file. At least one command in the HSIZIN file resulted in an error.

In the message text:

csid
current system identifier.

System action: Processing terminates.

Operator response: Examine the output in the HSIZMSG file to determine the problem(s).

System programmer response: None.

Module: HSIZMON

HSIZ062I *csid* **MAXCAD=***val* **IS PROBABLY TOO SMALL**

Explanation: A DSPSERV CREATE macro issued a return code of 12. This is usually caused by the maximum number of SCOPE=COMMON data spaces already existing, so that no more can be created. To increase this maximum specify a larger value for MAXCAD in the system parameter library for the next IPL.

In the message text:

csid
current system identifier.

val
current value of MAXCAD.

System action: Processing terminates.

Operator response: Restart the Usage Monitor after a SCOPE= COMMON data space has been deleted.

System programmer response: Allow a greater number of concurrent SCOPE=COMMON data spaces by increasing MAXCAD in PARMLIB.

Module: HSIZMON

HSIZ063I *csid* **COLLECTING "UNKNOWN" EVENTS?** *ans*

Explanation: Either a UNK command was issued or a D-S command was issued. When the answer is YES this message indicates that the Usage Monitor logs events with incomplete data which would not normally be collected. Data base content is not affected.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ064I *csid* **WILL WRITER TASK COMPRESS THE DATA?** *ans*

Explanation: Either a ZIP command was issued to change the output compression setting or a D-S command was issued. When the answer is YES the writer task writes compressed data to reduce I/O volumes.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ065I *csid* **WILL WRITER TASK CORRECT LINKLIST DSN?** *ans*

Explanation: Either an LLC command was issued or a D-S command was issued. When the answer is YES the writer task will perform a BLDL for programs known to have been fetched from the link list, and each output record for such programs will be altered to reflect the link list data set name that the writer task found the program in.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ066I *csid nbr* **IDLE ELEMENT(S) "LOST" DUE TO ZERO POINTER**

Explanation: The Usage Monitor was terminating normally when a storage accounting discrepancy was discovered. The storage for the idle element chain was being freed when it was found to be terminated by a zero pointer before the expected number of elements had been processed. The most probable cause is a storage overlay. This may or may not represent a Usage Monitor logic error. The size of common storage which may be unusable until the next IPL can be calculated by multiplying the element count by the size of an element.

In the message text:

csid
current system identifier.

nbr
the number of elements being reported.

System action: Termination continues.

Operator response: Determine if the size of the potential loss of common storage is likely to impact upon system stability, and take the appropriate action. Ensure that all appropriate maintenance has been applied.

System programmer response: None.

Module: HSIZMON

HSIZ067I *csid* **SAVE ELEMENTS ON THE IDLE CHAIN?** *ans*

Explanation: Either an IDL command was issued or a D-S command was issued. When the answer is YES the Usage Monitor will place processed work elements on a chain for idle elements instead of freeing the storage. When an address space needs an element to record a program usage event, one from the idle chain will be used in preference to acquiring more storage. Use of the idle chain can reduce GETMAIN/FREEMAIN processing and therefore improve overall efficiency.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ068I *csid* **COLLECTING JOB ACCOUNTS NOW?** *ans*

Explanation: A D-S command was issued. When the answer is YES job account data is currently being collected as program usage events are recorded. When the answer is NO job account data is not being collected currently.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ069I *csid* **COLLECTING JOB ACCOUNTS LATER?** *ans*

Explanation: Either a JAC command was issued or a D-S command was issued. When the answer is YES job account data will be collected after the next Usage Monitor collection repository switch. If the answer is NO job account data will not be collected from that time onwards.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ070I *csid* **COLLECTING REGISTERED PRODUCT DATA NOW?** *ans*

Explanation: A D-S command was issued. When the answer is YES registered software product data from SMF is currently being collected by the Usage Monitor. When the answer is NO then this SMF data is not being currently collected.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ071I *csid* **COLLECTING REGISTERED PRODUCT DATA LATER?** *ans*

Explanation: Either a PRS command was issued or a D-S command was issued. When the answer is YES registered software product data from SMF will be collected after the next Usage Monitor collection repository switch. When the answer is NO this SMF data will not be collected after the next switch.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ072I *csid* **COLLECTING DYNAMIC CAPACITY DATA NOW?** *ans*

Explanation: A D-S command was issued. When the answer is YES hardware capacity information is currently being collected by the Usage Monitor. When the answer is NO hardware capacity information is not being currently collected.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ073I *csid* **COLLECTING DYNAMIC CAPACITY DATA LATER?** *ans*

Explanation: Either a CAP command was issued or a D-S command was issued. When the answer is YES the Usage Monitor will collect hardware capacity information after the next Usage Monitor collection repository switch. When the answer is NO the hardware capacity information will not be collected after the next switch.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ074I *csid* **OUTPUT NAMES OF COLLECTED USERS?** *ans*

Explanation: Either a UNM command was issued or a D-S command was issued. When the answer is YES collected user names will be included in the data output by the Usage Monitor writer task. When the answer is NO user names will not be written to the output data set. Even if the answer is YES, no user names will be output if no user information was collected.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ075I *csid* **COLLECTING USER INFORMATION NOW?** *ans*

Explanation: A D-S command was issued. When the answer is YES the identifier and name of each program user is currently being collected by the Usage Monitor. When the answer is NO these user details are not being currently collected.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ076I *csid* **COLLECTING USER INFORMATION LATER?** *ans*

Explanation: Either a UID command was issued or a D-S command was issued. When the answer is YES the identifier and name of each program user will be collected after the next Usage Monitor collection repository switch. When the answer is NO these userid details will not be collected after the next switch.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ077I *csid* **COLLECTING JOB NAMES NOW?** *ans*

Explanation: A D-S command was issued. When the answer is YES the names of jobs using programs are currently being collected by the Usage Monitor. When the answer is NO only generic address space type data such as JOB, STC and TSU is currently being collected instead of individual job names.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ078I *csid* **COLLECTING JOB NAMES LATER?** *ans*

Explanation: Either a JNM command was issued or a D-S command was issued. When the answer is YES the names of jobs using programs will be collected after the next Usage Monitor collection repository switch. When the answer is NO only generic address space type data such as JOB, STC and TSU will be collected after the next switch instead of individual job names.

In the message text:

csid
current system identifier.

ans
YES or NO.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZMON

HSIZ080I *csid dsn*

Explanation: Displays the dataset name mask for a D-D command.

In the message text:

csid
current system identifier.

dsn
data set name.

System action: Processing continues.

Operator response: None required.

System programmer response: None.

Module: HSIZMON

HSIZ201I **DYNALLOC FAILURE RC=***rc* **ERROR=***s99error* **INFO=***s99info* **DSN=***dsn*

Explanation: The writer task could not dynamically allocate a new output data set.

In the message text:

rc DYNALLOC return code.

s99error
dynamic allocation reason code (DARC).

s99info
dynamic allocation information code.

dsn
name of the data set being allocated.

System action: Processing of the repository is terminated, and the data lost.

Operator response: Correct the cause of the allocation failure. If necessary, use the DSN, PRI, SEC and UNT commands to customize the allocation request for your installation. Note: The meanings of most DARC values are usually available in Appendix A of the ISPF Tutorial.

System programmer response: None.

Module: HSIZ0203

HSIZ202I USAGE MONITOR - COMPRESSION SUBROUTINE ERROR

Explanation: While processing repository data the compression subroutine encountered an error. The error message from the compression subroutine immediately follows this message.

System action: Processing of the repository is terminated, and the data lost.

Operator response: Correct the error described in the message from the compression subroutine. If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

System programmer response: None.

Module: HSIZ0203

HSIZ203I USAGE MONITOR - SORT FAILED - RC=*rc*

Explanation: While sorting repository data the SORT task ended with a non-zero condition code which is taken to mean that the sort was not successful. This message is followed by message HSIZ205I.

In the message text:

rc decimal return code of the sort subtask.

System action: The output data set is closed, and the writing of unsorted data to the same data set is attempted.

Operator response: Consult the documentation of the SORT utility. The contents of the SORT report file (DDNAME=SYSOUT) may be helpful.

System programmer response: None.

Module: HSIZ0203

HSIZ204I USAGE MONITOR - SORT ABENDED - ABEND CODE=*abend*

Explanation: While sorting repository data the SORT task ended abnormally. This message is followed by message HSIZ205I.

In the message text:

abend
the abend code of the sort subtask.

System action: The output data set is closed, and the writing of unsorted data to the same data set is attempted.

Operator response: Investigate why such an abend could occur. The contents of the SORT report file (DDNAME=SYSOUT) may be helpful.

System programmer response: None.

Module: HSIZ0203

HSIZ205I USAGE MONITOR - UNSORTED DATA WILL BE WRITTEN

Explanation: The sorting of output data has failed so the data is now written unsorted.

System action: The message is preceded by either HSIZ203I or HSIZ204I. After the SORT task ended the output data set has been closed and reopened. Repository data is about to be written to the output data set.

Operator response: Investigate why the sort failed.

System programmer response: None.

Module: HSIZ0203

HSIZ206I *errmsg*

Explanation: The HSISHRNK compression routine issued an error message which is displayed.

In the message text:

errmsg
error message from HSISHRNK.

System action: The message is preceded by message HSIZ202I.

Operator response: Examine the message for further information.

System programmer response: None.

Module: HSIZ0203

HSIZ301I DESERV FUNC=EXIT RC=*rc* REASON=*rs*

Explanation: DESERV FUNC=EXIT issued a non-zero return code.

In the message text:

rc return code from DESERV.

rs reason code from DESERV.

System action: The DESERV exit is not installed.

Operator response: Notify the system programmer.

System programmer response: Research the DESERV feedback to determine why the exit could not be installed.

Module: HSIZ0303

HSIZ302I CSVDYNEX ADD (*excd*) RC=*rc* REASON=*rs*

Explanation: CSVDYNEX ADD issued a non-zero return code. Exit differentiation code of '1' means that exit SYS.IEFU84 could not be installed, Exit differentiation code of '2' means that exit SYSSTC.IEFU84 could not be installed.

In the message text:

excd
exit differentiation code

rc return code from CSVDYNEX.

rs reason code from CSVDYNEX.

System action: The SMF exit is not installed.

Operator response: Notify the system programmer.

System programmer response: Research the CSVDYNEX feedback to determine why the exit could not be installed. If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIZ0303

HSIZ303I ATTRIBUTE MISMATCH - *mod* NOT INSTALLED

Explanation: The examined SVC table entry did not have the expected attributes.

In the message text:

mod
module name.

System action: The SVC intercept is not installed.

Operator response: Notify the system programmer.

System programmer response: Gather appropriate diagnostic materials and contact IBM support.

Module: HSIZ0303

HSIZ306I BAD *statnm* ENTRY PGM=*pgm* JOB=*jbn* USER=*user* ID=*id* DATE=*date* REJECTED

Explanation: An invalid work element has been detected and some of its contents are displayed.

In the message text:

statnm
status name.

pgm
program name.

jbn
job name.

user
user name.

id id name.

date
date.

System action: Attempted to dump some data to HSIZSNAP if the file is allocated, and will then try to free the work element without processing its contents.

Operator response: Notify the system programmer.

System programmer response: The problem is indicative of a storage overlay. Gather appropriate diagnostic materials and contact IBM support.

Module: HSIZ3060

HSIZ310I MODULE *mod* INSTALLED AT ADDRESS *loadpt* SIZE *size*

Explanation: The Usage Monitor has dynamically loaded a module into common storage and will now register it in DLPA.

In the message text:

mod
module name.

loadpt
module load point.

size
module size.

System action: Processing continues.

Operator response: None.

System programmer response: None.

Module: HSIZ0303

HSIZ311I CSVDYLPA RC=*rc* RS=*rs* FOR *mod*

Explanation: The Usage Monitor attempted to register a newly installed module in DLPA, but CSVDYLPA issued a non-zero return code.

In the message text:

rc decimal return code issued by CSVDYLPA.

rs hexadecimal reason code issued by CSVDYLPA.

mod
name of the module being registered.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Investigate why the named module could not be registered in the current DLPA configuration.

Module: HSIZ0303

HSIZ999U HSIMSG/HSIZMSG FAILURE - MSGID=*msgid* RC=*rc* RS=*rs*

Explanation: HSIMSG was called to produce a message text, but the call failed.

In the message text:

msgid
identifier of the failing message.

rc HSIMSG return code.

rs HSIMSG reason code.

System action: Terminates with a condition code of 20.

Operator response: Inform the system programmer.

System programmer response: Ensure Joblib/Steplib contains the library where the HSIZMSG message module resides. If you cannot resolve this issue then gather appropriate diagnostic materials and contact IBM support.

Module: HSIZMON :emsgl.

HSIC - Operation messages

Return codes

0	No errors encountered. All requests processed successfully.
16	Unrecoverable error. No requests processed. SYSIN or HSIPZIP or INQSOUT File cannot be used, or unsupported operating system.

Message suffix codes

Suffix	Meaning	Raises Minimum Cond Code to
I	Information Message	0
W	Warning Message	4
E	Error Message	8
S	Severe Error Message	12
U	Unrecoverable Error Message	16

Message texts and explanations

All numeric completion codes of system services reported in these messages are in hexadecimal unless otherwise stated.

HSIC002E A message is missing from the internal repository

Explanation: A message is missing from the internal message repository. When the default language is not English, it could simply mean that no translation of the given message exists. If the default language is English, that would indicate an error in the given application.

System action: The application would normally continue ignoring the given message number , but the specific action depends on the code attempting to issue the message which could also terminate the application.

User response: Contact IBM support.

HSIC003U The internal message repository is corrupted

Explanation: When attempting to issue a message, the internal message repository layout did not follow the expected format.

System action: The application terminates.

User response: Contact IBM support.

HSIC020E *application-name* encountered errors. Error code = *errorcode*

Explanation: The Application has encountered errors during processing. This is a general message on

completion indicating that an error has occurred.

System action: Completes with given error code .

User response: Refer to additional message, or to the section "Return codes" on page 162, and to the log for more details on the specific error. Contact IBM support.

HSIC021S *application-name* **encountered fatal errors. Error code =** *error-code*

Explanation: The Application has encountered fatal errors during processing.

System action: Terminates with given error code

User response: Refer to additional message, or to the section "Return codes" on page 162, and to the log for more details on the specific error. Contact IBM support.

HSIC023E **Inquisitor Import error occurred in opening:** *filename*

Explanation: The Inquisitor import could not open the given file.

System action: Terminates without processing any records.

User response: Check that the file exists, and if it does, check for any additional log message identifying the error. Contact IBM support.

HSIC024E **Inquisitor Import input file is in error. It looks like a usage data file**

Explanation: The inquisitor import has encountered an invalid input file.

System action: Terminates without processing any records.

User response: Check that the input file is a valid file. Contact IBM support.

HSIC025E **Inquisitor Import input file is in error. It looks like a hardware data file**

Explanation: The Inquisitor Import has encountered an invalid input file.

System action: Terminates without processing any records.

User response: Check that the input file is a valid file. Contact IBM support.

HSIC026E **Inquisitor Import detected that table** *tablename* **is missing or invalid**

Explanation: The expected table is missing from the database or has invalid format. This suggests a mismatch between the database and this version of the product.

System action: Terminates without processing any records.

User response: Check for a version mismatch between the database and the version of the product. Contact IBM support.

HSIC027S **Inquisitor Import table** *tablename* **is missing a column**

Explanation: The given table is missing an expected column. This suggests a mismatch between the database and this version of the product.

System action: The application terminates without processing any records.

User response: Check for a version mismatch between the database and the version of the product. Contact IBM support.

HSIC028S **Inquisitor Import table** *tablename* **appears to be an old version**

Explanation: The given table in the database does not have the expected format.

System action: The application terminates without processing any records.

User response: Check for a version mismatch between the database and the version of the product. Contact IBM support.

HSIC029S **Inquisitor Import error when writing to table** *tablename*

Explanation: An SQL error occurred when attempting to write to the given table.

System action: The application terminates.

User response: Check the log for additional details about the given error. Contact IBM support.

HSIC030S **The Inquisitor Import did not find a valid system header record in the input file**

Explanation: The input file does not follow the expected format.

System action: The application terminates.

User response: Check that the correct input file is supplied, and that there is no version mismatch. Contact IBM support.

HSIC034S **Error reading Repository TPARAM table**

Explanation: An error occurred while reading the TPARAM Repository table.

System action: The application terminates.

User response: Check the log for any additional

messages indicating the cause of the error. Contact IBM support.

HSIC035E **The Repository is in use by the** *application-name*

Explanation: The application cannot run because the Repository is already in use by another application. Wait until *application-name* completes before running the current application. If the Repository is not in use by *application-name*, then the cause could be that it was previously run, but did not run to completion. To correct the problem, either rerun the *application-name* identified in this message, or alternatively, run the HSISTPRM supplied job to reset FVALUE to 0 where FKEY = PROCRUN in the TPARAM table.

System action: The application terminates.

User response: Check the application is not already in use, before running this application.

HSIC036E **Syntax error scanning TPARAMS file on line** *linenumber*

Explanation: The TPARAM file does not conform to the required syntax on the given line.

System action: The specified option or value is ignored, and its default value is used where applicable.

User response: Check that valid options/values are supplied as specified in the documentation of the application that you are running.

HSIC037E **Schema** *schemavalue* **is too long in param** *param*

Explanation: A schema id that is too long has been specified.

System action: The application terminates.

User response: Check that the schema id does not exceed 8 characters in length.

HSIC038E **Unbalanced quote for value:** *value* **in param:** *param*

Explanation: A starting quote was found for the given parameter that has no matching end quote.

System action: The application terminates.

User response: Check that the given parameter has matching quotes

---

HSIC039E **Illegal character in value**:*value* **of param:***param*

Explanation: An invalid character was found in the given value.

System action: The application terminates.

User response: Check that the given parameter value is valid for its type.

HSIC040E **Reserved word:** *reservedword* **in param:** *param*

Explanation: A reserved word or system value schema ID was chosen as a parameter value.

System action: The application terminates.

User response: Specify a different parameter value

HSIC041W **value:***value* **in param:***param* **is not a recommended schema ID**

Explanation: The value is not recommended because of possible conflicts with existing values.

System action: The application continues.

User response: Please choose a different value to avoid any conflicts

---

HSIC042E **TPARAM file: param**:*param* **has an invalid proposed value:** *value*

Explanation: The parameter cannot be set to the given value, because the value is not valid..

System action: The value is ignored, and the application continues.

User response: Please choose a valid value as per the documentation of the given application

---

HSIC043E **The application has failed to open the TPARAM file. Error:** *errordescription*

Explanation: The application could not open the TPARAM file. The error description contains more details regarding the reason for the error.

System action: The application terminates.

User response: Check that the TPARAM file exists and is valid.

HSIC045E **String** *string* **cannot exceed** *numberchars* **in length**

Explanation: A parameter length limit has been exceeded.

System action: The application terminates.

User response: Ensure that the specified parameter length is not exceeded.

HSIC050E **The** *program-name* **program has detected an invalid date parameter**

Explanation: A date parameter was found to be invalid.

System action: The application terminates.

User response: Ensure that the date format is valid, and start dates do not overlap end dates.

HSIC051S Error adding record

Explanation: An SQL error occurred when adding a record to a table.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC052S Error updating record

Explanation: An SQL error occurred when updating a record in a table.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC053S Error deleting record

Explanation: An SQL error occurred when deleting a record from a table.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC054E Usage Summary detected an invalid SUMBY value

Explanation: The Usage Summary detected an invalid SUMBY value.

System action: The specified value is ignored. The application continues using the default SUMBY value.

User response: Refer to the documentation of the Usage Summary parameter for valid SUMBY values.

HSIC055S Table initialization failure during Repository Merge

Explanation: At least one table initialization failed when merging repositories.

System action: The application terminates.

User response: Check the log for any additional details about this error. Contact IBM support.

HSIC056S Some table destination fields are smaller than source

Explanation: Some fields in the target repository are not large enough to fit the contents of fields in the source repository.

System action: The application terminates, and the repositories are not merged.

User response: Check that the destination repository is not an older version than the source repository. You can recreate the destination repository using the latest version of the product. If the problem persists, contact IBM support.

HSIC057E A value for parameter: *parameter-name* **must be specified**

Explanation: A mandatory parameter for this application has not been specified.

System action: The application terminates during the syntax checking of input parameters.

User response: Ensure that a value for the given parameter is specified. Refer to the documentation of the failing application for an explanation of the given parameter and/or valid parameter values.

HSIC058E Could not open *filename*

Explanation: File could not be opened.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC059E Could not read *filename*

Explanation: File could not be read.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC060E IQDATA DD does not contain unzipped IQ data

Explanation: The input IQDATA dataset does not contain unzipped IQ data.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC061E Internal error hcreate*(number)* **phase1a failed**

Explanation: An internal error has occurred.

System action: The application terminates

User response: Check the log for additional information about the error. Contact IBM support.

HSIC062E No SMF 30-2 or 30-4 data matched IQ data

Explanation: No match was found for the SMF data and IQ data.

System action: The application terminates.

User response: Check that the correct data sets have been used. Contact IBM support

---

HSIC063E Internal error hsearch*(key)* table add failed

Explanation: An error occurred when inserting data into a table.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC064E Could not write *type* to FMOUT

Explanation: Could not write to file FMOUT.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC065E SYSUT1 data Is not IQ text or UM text

Explanation: The SYSUT1 dataset does not contain the expected data.

System action: The application terminates.

User response: Check that the SYSUT1 dataset is correct. Contact IBM support.

HSIC066E Internal error hsearch*(key)* table failed

Explanation: An error occurred when retrieving data from a table.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC067E Unable to acquire storage

Explanation: An error has occurred when attempting to acquire storage.

System action: The application terminates.

User response: Try increasing the region size specified in the region parameter on the JOB or EXEC statement in the JCL for the job. Contact IBM support.

HSIC068E IBMMOD Internal error

Explanation: An internal error has occurred.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC069E IBMMOD_INIT internal error

Explanation: An error occurred when retrieving data from a table.

System action: The application terminates.

User response: Check the log for additional information about the error. Contact IBM support.

HSIC070I A full rematch will be performed

Explanation: A full import and rematch will be performed, which will not try to exclude modules of unchanged libraries. The default behaviour is to exclude such libraries from matching, which would normally lead to faster processing.

The program performs a full rematch, if any of the following is true :

- If requested by the FULLREMATCH option.
- When the specified inventory is not found, for example on the first run when the inventory has not yet been created, and no previous match was done.
- If it is safer to perform a full rematch, as when a GKB change is detected or the REPLACEFULL option is in effect.

More specific details on why a full rematch is being performed, can be found in the log.

System action: A full rematch of the data is performed. All libraries are processed.

User response: Ensure that a FULLREMATCH and the REPLACEFULL options are not in effect for better performance, unless a full rematch is desired.

If this is the first run of the Inquisitor Import, or there has been recent a change to the GKB, then no action is necessary; the program will try on subsequent runs (subsequent to loading the current data into the repository) to exclude unchanged libraries.

HSIC071I *&number_modules* modules in *&number_libraries* unchanged libraries were ignored

Explanation: This is a report of the number of the modules and libraries that are ignored when the FULLREMATCH option is not in effect. Details of these ignored libraries are in the log.

System action: None.

User response: None.

Return codes

6016	Input text file open error
6060	Input Parameter error
6061	Database open error
6062	Database commit error
6063	Error reading repository TPARAM table
6065	Repository is in use
6066	Unknown SID parameter value
6203	Inquisitor Import table open fail
6204	MVS system header record not found in input file
6205	Unix System Services header record not found in input file
6206	No system header record found in input file
6208	Error writing to TPARAM table
6209	Error opening input file
6211	Fatal error writing system record
6212	Fatal error writing library record
6213	Fatal error writing module record
6218	Input file looks like a usage data file
6219	Input file looks like a hardware data file
6220	Index missing error
6221	Vendor product version table processing error
6222	Tagged module key table processing error
6223	Error encountered when retrieving the inventory ID
6224	Error encountered when retrieving the current GKB version
6225	Error encountered when retrieving the inventory GKB version
6237	Inquisitor Import table does not exist or is a missing a column
6238	Inquisitor Import table does not exist
6239	Inquisitor Import table appears to be an old version
6240	Error updating fGPassLibID record
6241	Error deleting empty libraries
6244	Error assigning package information to TMODULE records
6260	Nothing to import, as no module records were found in IQ file
6400	Knowledge Base type is incorrect
6402	Failure in initializing IQ tables
6403	IQ TMODULE open error
6404	IQ TMODULE index error
6405	IQ database is empty
6409	TDECISION table open error
6413	Error creating scorecard tables for Match Engine
6428	Local KB TRULES table open error

6434	Failure to open archive file
6435	Error creating index
6436	Error setting current index
6437	Search KB phase error
6438	Volume serial library phase error
6439	Inter Library phase error
6440	Rules processing phase error
6444	LPA phase error
6448	Error while clearing LMOD count
6449	TDECISION Table is missing FDECRPTION and/or FCATEGORY fields
6450	GKB TPRODUCT record seek error
6451	LKB TPRODUCT record seek error
6452	TDECISION record edit error
6453	KB TVERSION record access error
6454	KB TPRODUCT record access error
6455	KB TVENDOR record access error
6600	Match Engine tables TDECISION and/or TMIGREPORT are missing
6619	Error opening TPACKAGE table
6620	Repository table initialization failed
6621	Failure opening IQ table
6622	Unable to access GKB TVERSION table
6623	IQ TMODULE table is empty
6624	Predecessor inventory ID key does not exist
6625	Repository is not enabled for Unix System Services
6626	Repository must be enabled for Unix System Services, when the REPLACE option is in effect
6627	SYSPLEX ID mismatch in inventory record
6628	SMFID mismatch in inventory record
6629	Inventory ID key of zero is not valid
6630	Error in deleting library record
6632	Error transferring TLIBRARY information from IQ to Repository
6633	Error accessing TINVCTL table
6634	Mismatch found between the TINVCTL record flag and the REPLACE option
6635	Error updating FMODCNT field in TLIBRARY and TPOVLIB tables
6636	Product version key error
6637	Module key error
6639	Error updating FINVID18 fields in TUIMPORTCTRL table
6640	Error updating FINVID field in TINVREG table
6641	Error updating FINVID field in TINVREG table
6642	Error updating summary tables

6643	Error querying table in FMODID order
6645	Error marking TLIBRARY, TMODULETPOVLIB and TPOVINV records as deleted
6647	Repository type does not match IQ type
6648	When using a Continuous Inventory, an Inventory Name must be specified
6666	Error when accessing the TLIBSYS table
6800	At least one repository failed during initialization
6802	No matching LPAR found in table
6803	Primary Inventory ID set to 0 for LPAR
6804	Error trying to find FMODID or FLIBID
6805	Inventory ID does not exist
6806	Unable to find or create TLPAR record for LPAR
6807	Error trying to find or create Job or User entry
6808	Error writing MTD record
6809	Error updating summary tables
6810	Error adding TUSELIBRARY record
6811	TLIBRARY update error
6812	Summary table error
6813	Error reading import control record
6814	User initiated stop
7000	At least one table failed initialization
7002	Invalid usage summary parameters
7003	Invalid month in usage summary parameter
7004	Date order error
7005	TMODULE record seek error
7011	Error inserting record into TMODULE table
7013	TJOBDATA record seek error
7014	TJOBDATA record add error
7015	TUSERDATA record seek error
7016	TUSERDATA record add error
7017	TUSEMTD record seek error
7018	TUSEMTD record add error
7019	TUSEMTD record edit error
7020	TUSEMTD record delete error
7021	TPOVINV record seek error
7022	TPERIODS record seek error
7023	TPERIODS record add error
7024	TPERIODS record edit error
7025	TUSEPOVLIB record seek error
7026	TUSEPOVLIB record add error
7027	TUSEPOVLIB record edit error
7028	TUSEPOV record seek error
7029	TUSEPOV record add error

7030	TUSEPOV record edit error
7034	TUSEMTD critical failure
7035	TUSEMTD error updating record with zero FMTDID
7036	TVERSION record seek error
7037	TUSEPO record seek error
7038	TUSEPO record seek error
7039	TUSEPO record edit error
7040	TUSEPO record delete error
7043	TMODULE record edit error
7044	TUSEPOVLIB record delete error
7045	TUSEPOV record delete error
7046	TPERIODS record delete error
7051	TUSELIB record delete error
7052	IDS_USUM_TUSELIB_ AUTONUM_ERROR
7055	TLPAR record edit error
7056	TUSELIB record seek error
7057	TUSELIB record add error
7058	TPOVLIB record seek error
7060	TLPAR record seek error
7061	Join record seek error
7062	TLIBRARY record edit error
7063	TLIBRARY record seek error
7065	Invalid SUMBY value
7066	Date formatting error
7067	Usage Summary schema is empty
7068	PRODUCT_USE delete error
7069	PRODUCT_USE_DETAIL delete error
7201	Inventory to be deleted does not exist in repository
7203	TLIBRARY record delete failure
7204	TPOVINV record delete failure
7205	TPERIODS record delete failure
7206	TLPAR record delete failure
7207	TUIMPCTRL record delete failure
7208	Failure updating Delete Inventory ID record
7209	Failure deleting TINVCTL records of deleted inventory
7210	Error scanning product version
7211	Error reassigning predecessor links in successor InvCTL records
7600	Table initialization failure
7601	Destination repository column size failure
7602	TINVCTL record seek error
7603	TINVCTL record edit error
7604	TINVCTL record add error

7605	TINVCTL record delete error
7606	TLIBRARY record seek error
7607	TLIBRARY record edit error
7608	TLIBRARY record add error
7609	TLIBRARY record delete error
7610	Transfer product version join seek error
7611	TPOVLIB record seek error
7612	TPOVLIB record edit error
7613	TPOVLIB record add error
7614	TPOVLIB record delete error
7615	TPOVINV record seek error
7616	TPOVINV record edit error
7617	TPOVINV record add error
7618	TPOVINV record delete error
7619	Table TINVPOV failed in initialization
7620	TVERSION record seek error
7621	TVERSION record edit error
7622	TVERSION record add error
7623	TVERSION record delete error
7624	Table TVERSION open failed
7625	TPRODUCT record seek error
7626	TPRODUCT record edit error
7627	TPRODUCT record add error
7628	TPRODUCT record delete error
7629	TPRODUCT open error
7630	TVENDOR record seek error
7631	TVENDOR record edit error
7632	TVENDOR record add error
7633	TVENDOR record delete error
7634	TVENDOR open error
7635	TMODULE record seek error
7636	TMODULE record edit error
7637	TMODULE record add error
7638	TMODULE record delete error
7639	TREGCLASS record seek error
7640	TREGCLASS record edit error
7641	TREGCLASS record add error
7642	TREGCLASS record delete error
7643	TREGION record seek error
7644	TREGION record edit error
7645	TREGION record add error
7646	TREGION record delete error
7647	TREGLEAF record seek error
7648	TREGLEAF record edit error

7649	TREGLEAF record add error
7650	TREGLEAF record delete error
7651	TINVREG record seek error
7652	TINVREG record edit error
7653	TINVREG record add error
7654	TINVREG record delete error
7655	TJOBDATA record seek error
7656	TJOBDATA record edit error
7657	TJOBDATA record add error
7658	TJOBDATA record delete error
7659	TUSERDATA record seek error
7660	TUSERDATA record edit error
7661	TUSERDATA record add error
7662	TUSERDATA record delete error
7663	TLPAR record seek error
7664	TLPAR record edit error
7665	TLPAR record add error
7666	TLPAR record delete error
7667	TUSEMTD record seek error
7668	TUSEMTD record edit error
7669	TUSEMTD record add error
7670	TUSEMTD record delete error
7671	TUSELIB record seek error
7672	TUSELIB record edit error
7673	TUSELIB record add error
7674	TUSELIB record delete error
7675	TPERIODS record seek error
7676	TPERIODS record edit error
7677	TPERIODS record add error
7678	TPERIODS record delete error
7679	TUSEPOVLIB record seek error
7680	TUSEPOVLIB record edit error
7681	TUSEPOVLIB record add error
7682	TUSEPOVLIB record delete error
7683	TUSEPOVLIB open error
7684	TUSEPOV record seek error
7685	TUSEPOV record edit error
7686	TUSEPOV record add error
7687	TUSEPOV record delete error
7688	TUSEPOV open error
7689	TUSEPO record seek error
7690	IDS_MRGE_TUSEPO_EDIT_ERROR
7691	TUSEPO record add error
7692	TUSEPO record delete error

7693	**TUSEPO open error**
7694	**TUIMPORTCTRL record seek error**
7695	**TUIMPORTCTRL record edit error**
7696	**TUIMPORTCTRL record add error 7697**
7697	**TUIMPORTCTRL record delete error**
7698	**Source and destination repositories are not the same type**
7699	**Source and/or Destination Repositories are not the correct category database**

Appendix B. Analyzer

This appendix contains information about:

Analyzer online mode navigation

The Analyzer online mode of navigation uses a PC Browser, for example Firefox, to communicate with the Analyzer for interactive queries.

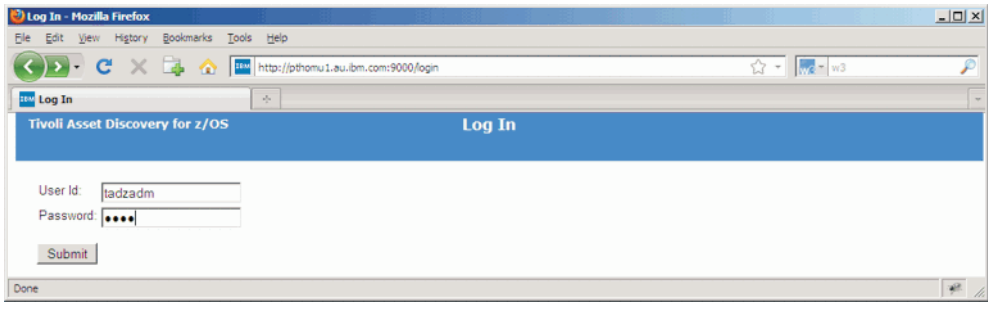
The screen prints in this appendix have a resolution of 1024 x 768 so that they can fit into this manual. However it is recommended that you use a larger screen resolution, such as 1440 x 900 or larger, as some of the reports contain a large amount of information.

If the Analyzer has been configured with `SECURITY=SYSTEM`, you communicate with HTTPS (HTTP with SSL encryption) and logon with your z/OS system user ID and password.

If the Analyzer has been configured with `SECURITY=BASIC` (the default), you communicate with HTTP and logon with a configured user ID and password. The user ID and passwords are supplied by default:

- `TADZADM / TADZ` This user ID has full access.
- `TADZUSR / TADZ` This user ID does not have access to, and does not see, the Administration menu items.

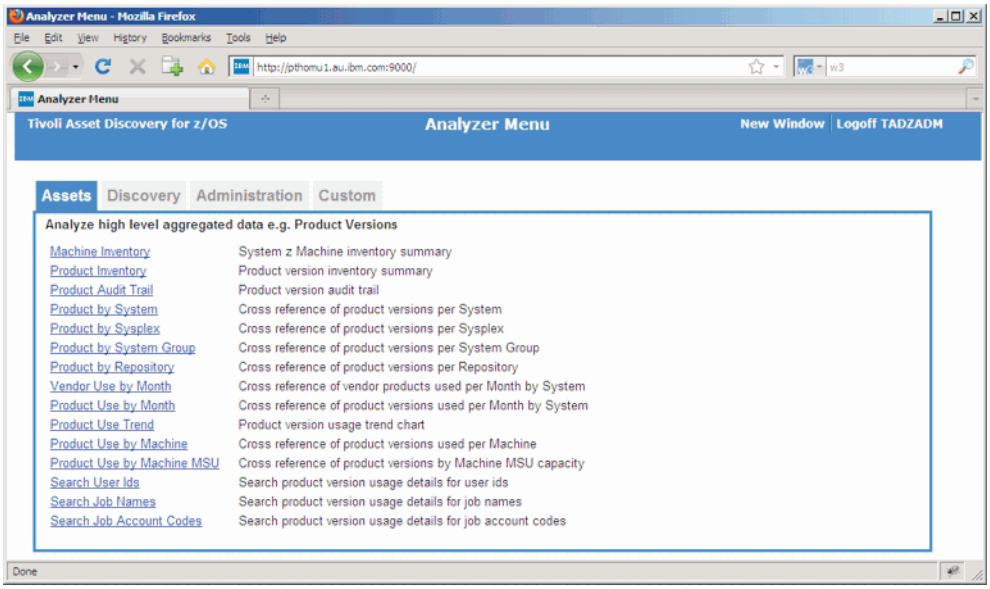
When you first connect to the Analyzer you are prompted to enter your user ID and password:



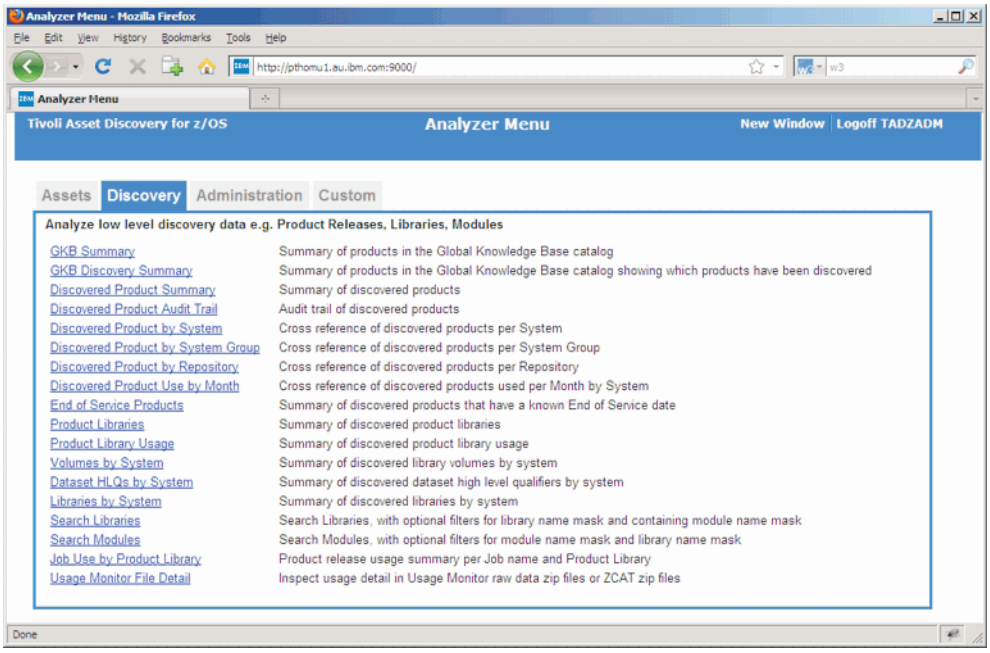
After you logon, you will see a screen with several menu tabs.

The **Assets** menu tab contains reports that query high level aggregated data; for example Product Versions. This level of data is useful if you are reconciling

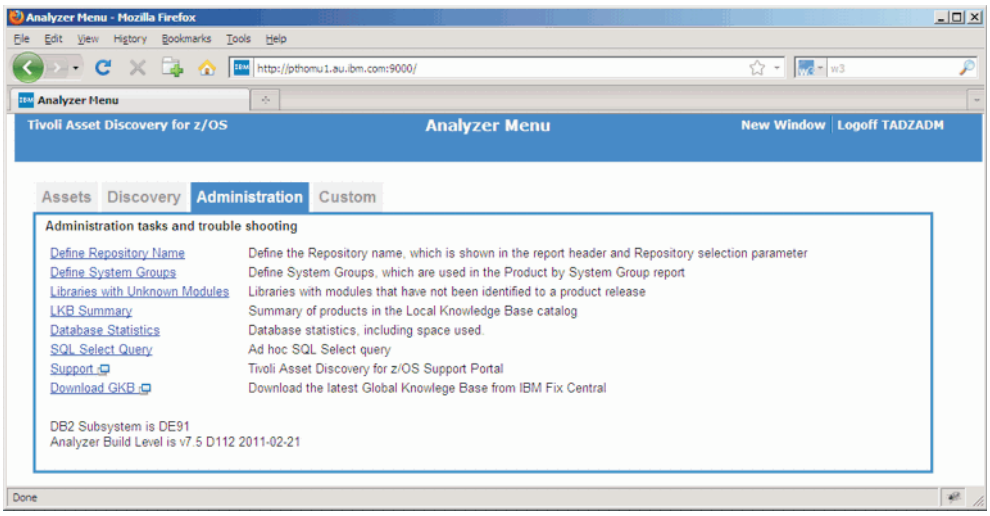
product licenses.



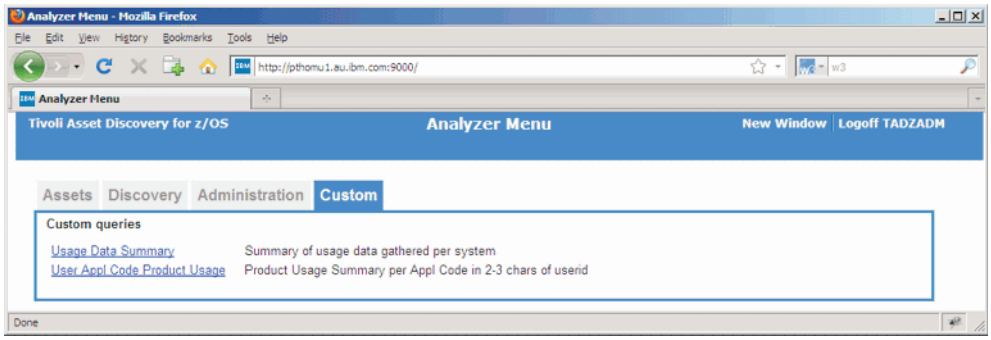
The **Discovery** menu tab contains reports that query low-level discovery data; for example, Product Releases, libraries, and modules. This level of data is useful if you support z/OS systems.



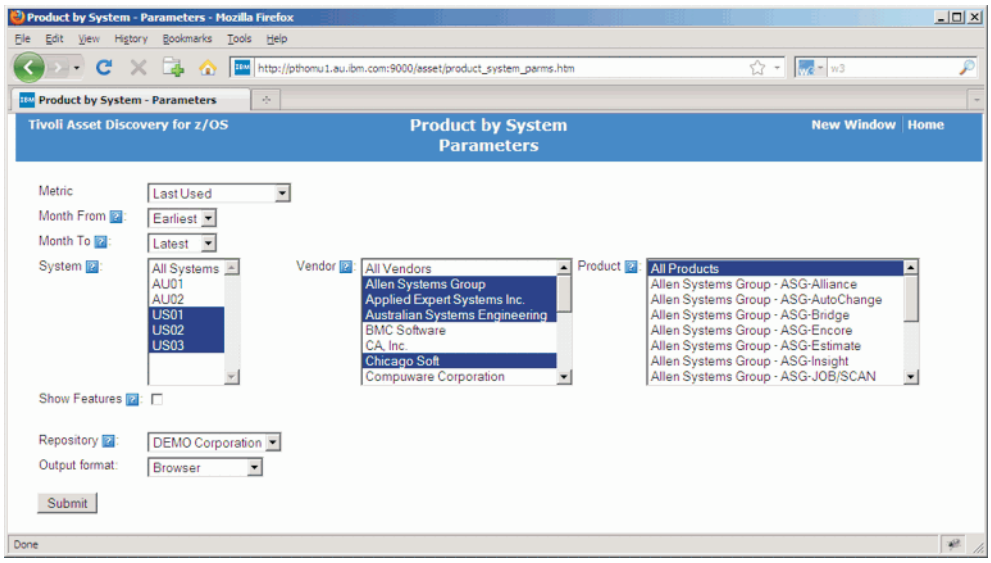
The **Administration** menu tab contains administration tasks and troubleshooting reports. These reports are designed for Tivoli Asset Discovery for z/OS administrators. Users do not see this menu unless they have been granted specific access.



The **Custom** menu tab contains reports that you have customized locally. Two samples are provided. See the "Analyzer custom queries" on page 84 section for more details.



When you click a report, you are presented with a parameter screen appropriate for that report. For example:

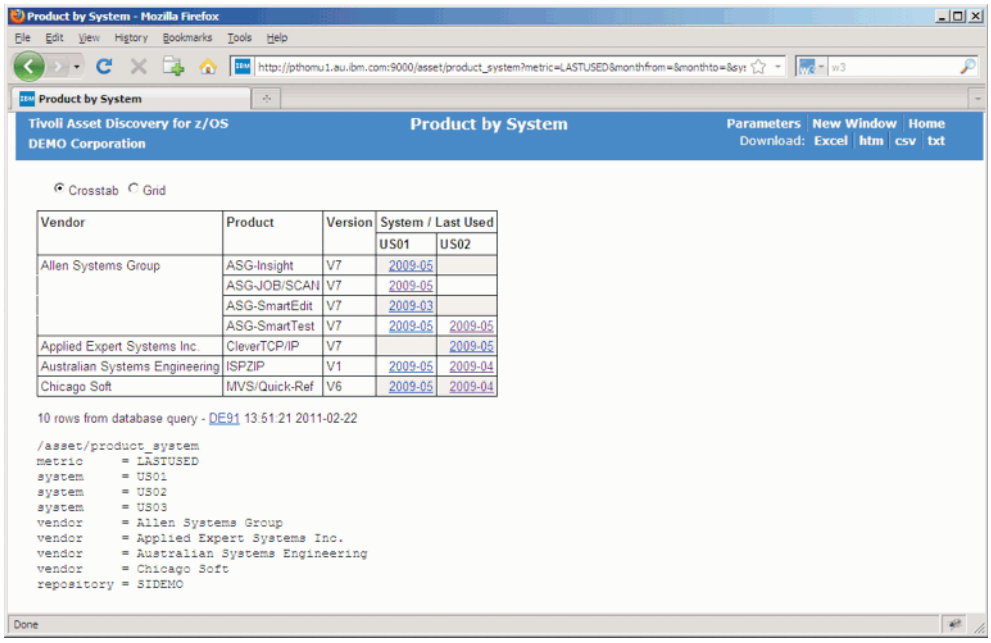


Parameter selection lists are based on the data that is in your database. The database in the preceding example contains data for five systems; AU01, AU02, US01, US02, and US03.

Multiple selections are possible by clicking the item you want while holding down the **Ctrl** or **Shift** key.

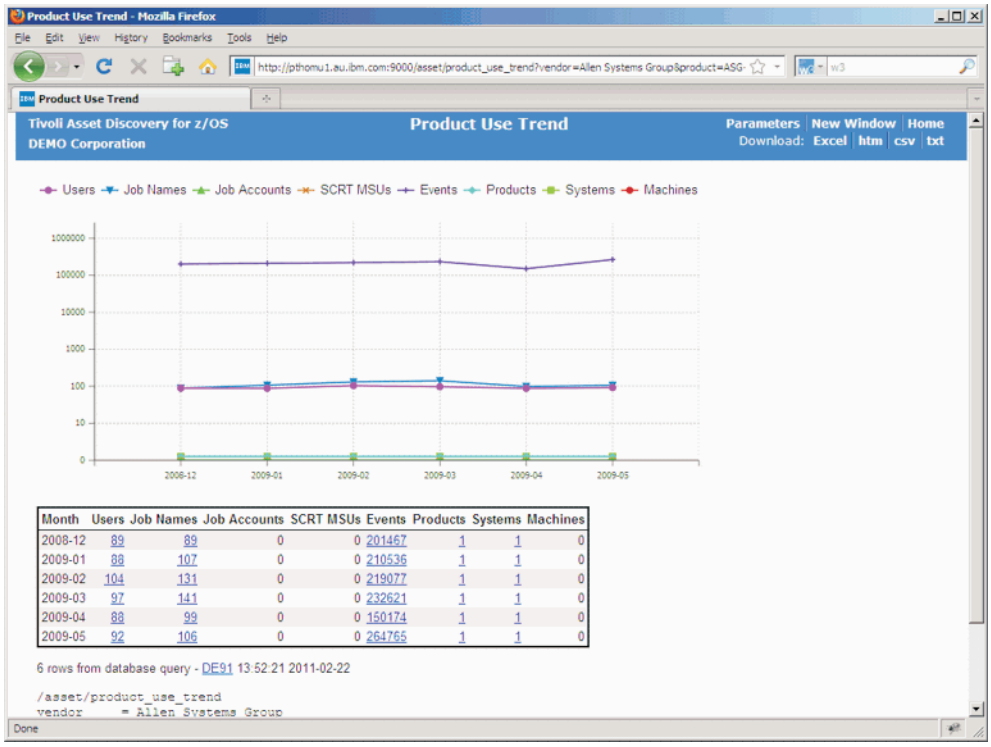
On the upper right of the screen there are hyperlinks to open a **New Window** or go to the **Home** menu screen.

Click **Submit** at the bottom of the screen to run the Analyzer query.

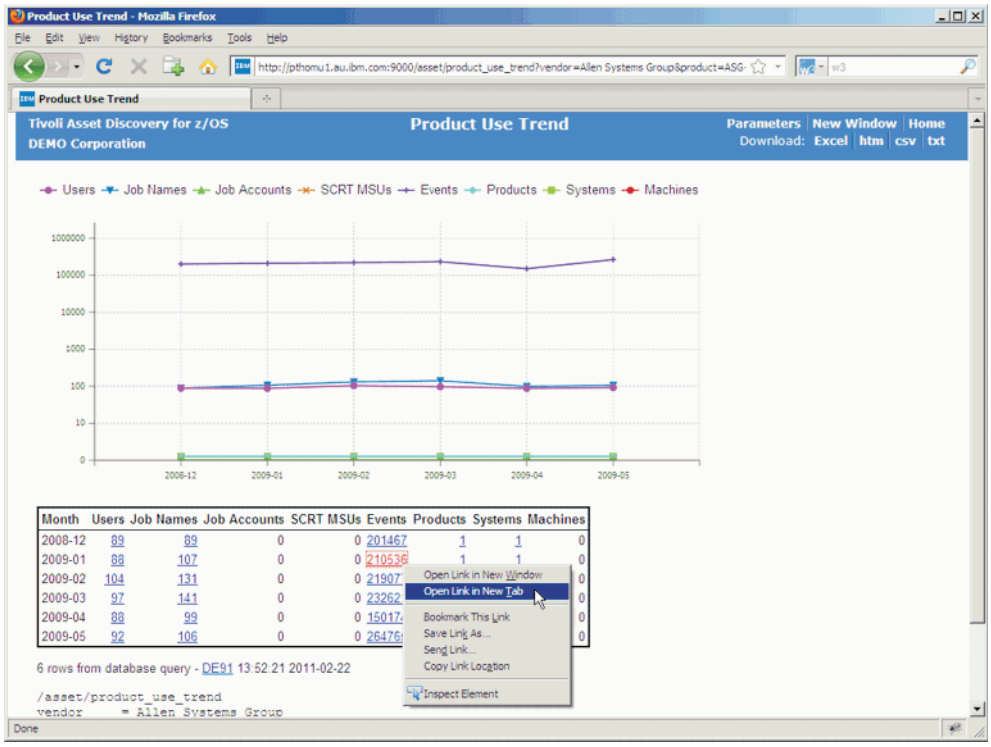


At the bottom of every report, the report name and parameters are shown in the same syntax that you can use to run the report in batch mode. Cut and paste from the report into the HSISANLB batch job SYSIN DD deck.

If you choose the Output format as Browser (the default), you get results that have hyperlinks that you can drill down on for more information. For example, clicking the **Last Used** hyperlink in the preceding sample invokes the **Produce Use Trend** report with the appropriate filter parameters.

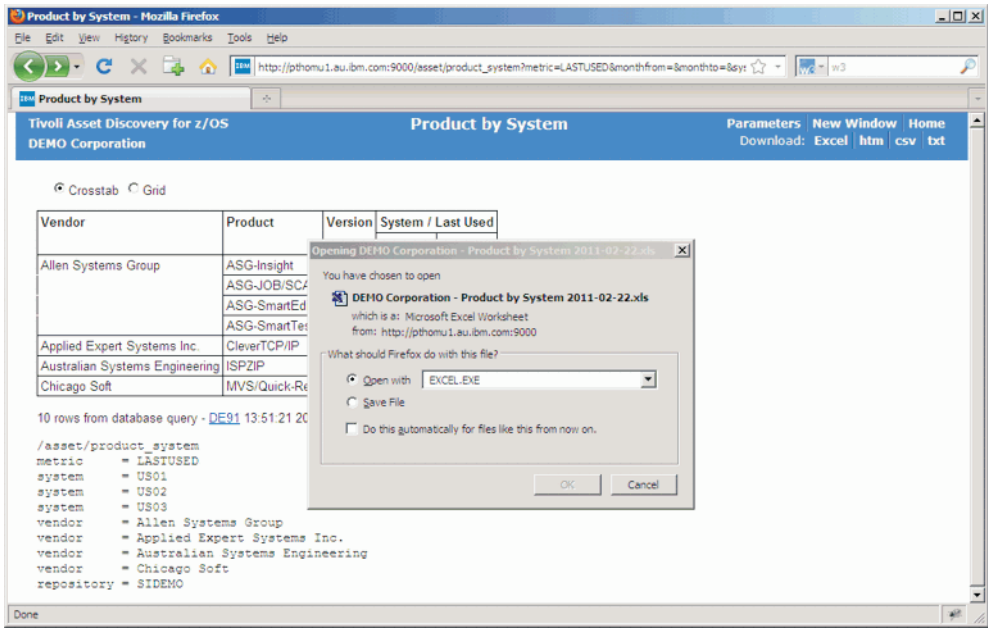


If you want to compare several drill downs, right click the hyperlink and select **Open Link in New Tab**.

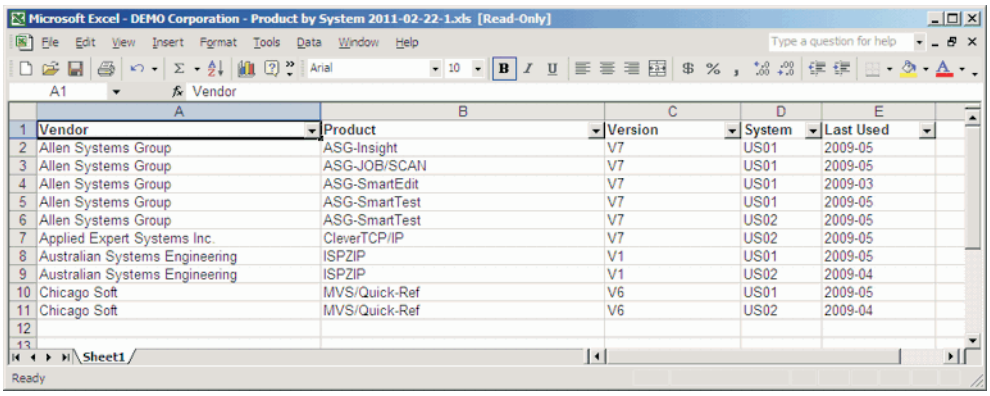


All HTML reports have embedded content so you can save them using the Browser menu; **File** and **Save**, or by using the Analyzer Download HTM and emailing them.

When you download the report in Excel, you are prompted to save the file or open in Excel. To save the file in Excel you need Microsoft Excel 2002, or above, installed on your PC. If you have a different spreadsheet package installed, choose the **Download: csv** option.



The Excel spreadsheet has a filter bar within the column headings for easy filtering.



	A	B	C	D	E
1	Vendor	Product	Version	System	Last Used
2	Allen Systems Group	ASG-Insight	V7	US01	2009-05
3	Allen Systems Group	ASG-JOB/SCAN	V7	US01	2009-05
4	Allen Systems Group	ASG-SmartEdit	V7	US01	2009-03
5	Allen Systems Group	ASG-SmartTest	V7	US01	2009-05
6	Allen Systems Group	ASG-SmartTest	V7	US02	2009-05
7	Applied Expert Systems Inc.	CleverTCP/IP	V7	US02	2009-05
8	Australian Systems Engineering	ISPZIP	V1	US01	2009-05
9	Australian Systems Engineering	ISPZIP	V1	US02	2009-04
10	Chicago Soft	MVS/Quick-Ref	V6	US01	2009-05
11	Chicago Soft	MVS/Quick-Ref	V6	US02	2009-04
12					

The **Download: htm** option saves the HTML report without hyperlinks. This report is useful if you want to email the reports.

The **Download: txt** option produces a simple text line mode file.

The **Download: csv** option produces a Comma Separated Value file. This format is compatible with other tools, including open source spreadsheet tools.

Analyzer Asset reports

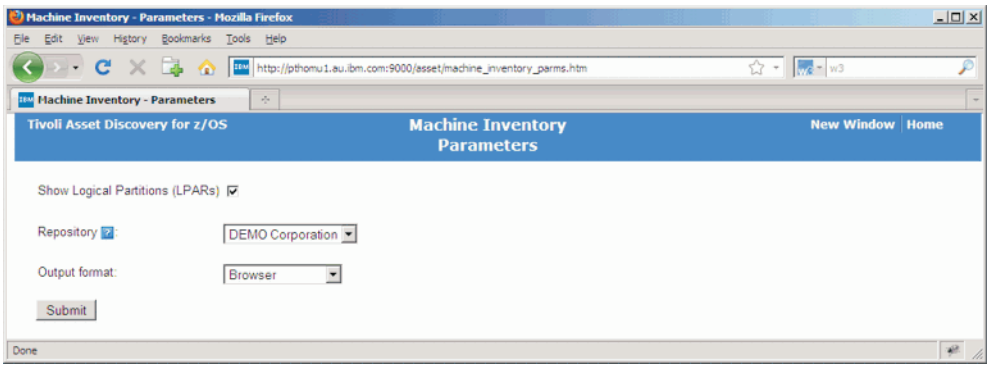
This section contains a list of the asset reports. For each report, two sample screens are provided. The first screen displays sample parameters, followed by sample output. Directly following these screens is a table which describes the function of each hyperlink in the output section of the report.

Machine Inventory

Description: System z® Machine inventory summary

Batch name:`/asset/machine_inventory`

Sample parameters:



Sample output:

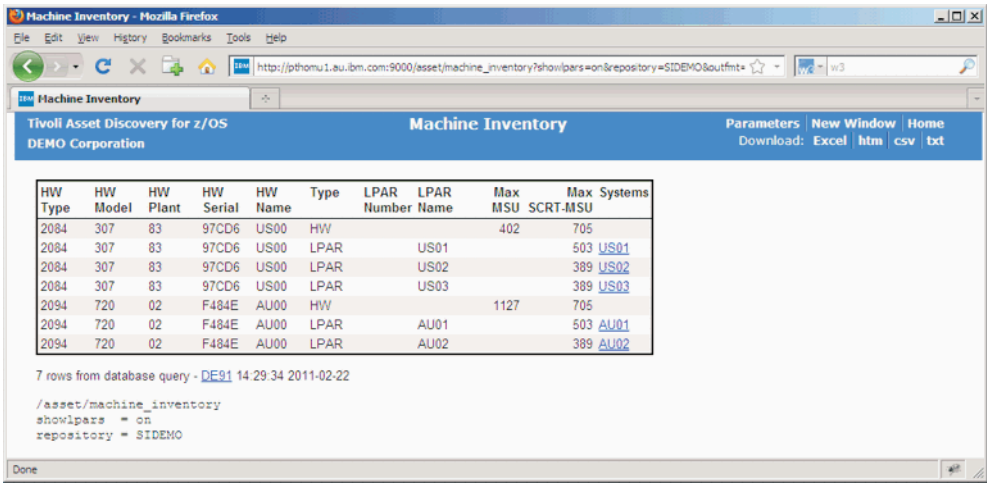


Table 26. Machine Inventory hyperlinks

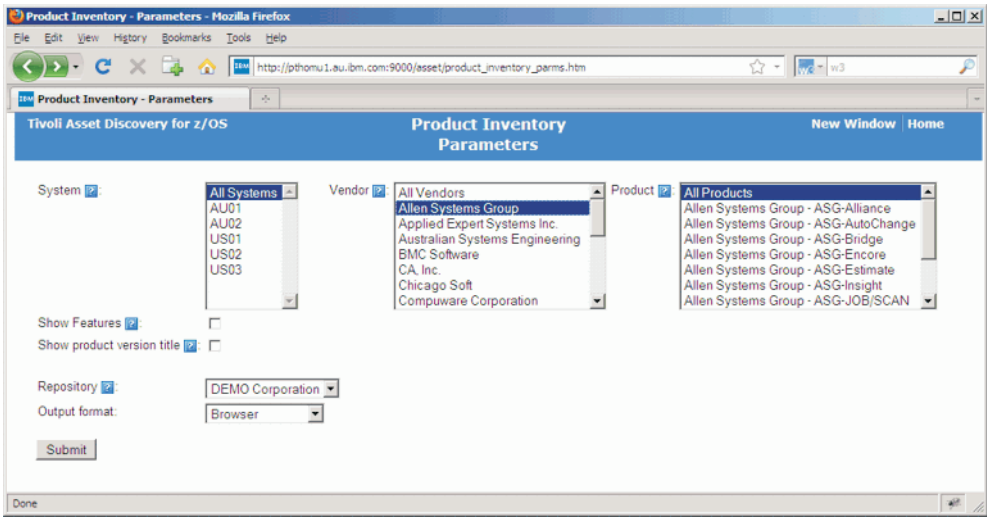
Column	Hyperlink
Systems	Drill down to **Product by System** report to see more information about products that have been used on Systems which have run on the Machine, or Logical Partition (LPAR).

Product Inventory

Description: Product version inventory summary

Batch name: /asset/product_inventory

Sample parameters:



Sample output:

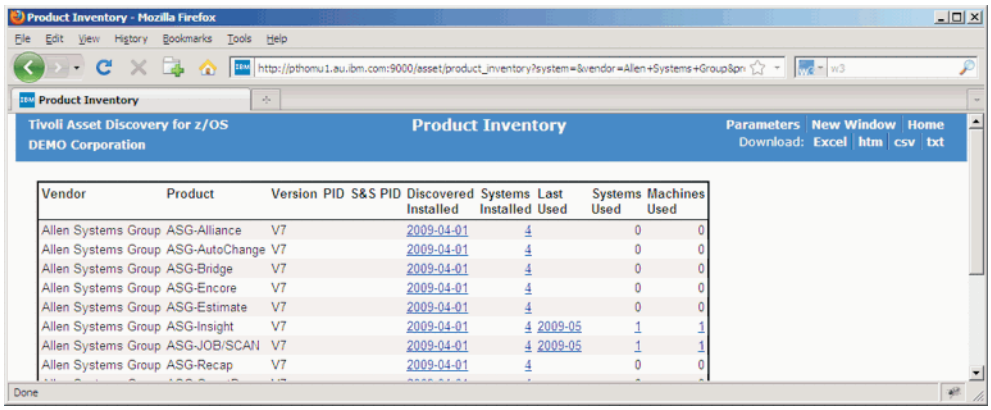


Table 27. Product Inventory hyperlinks

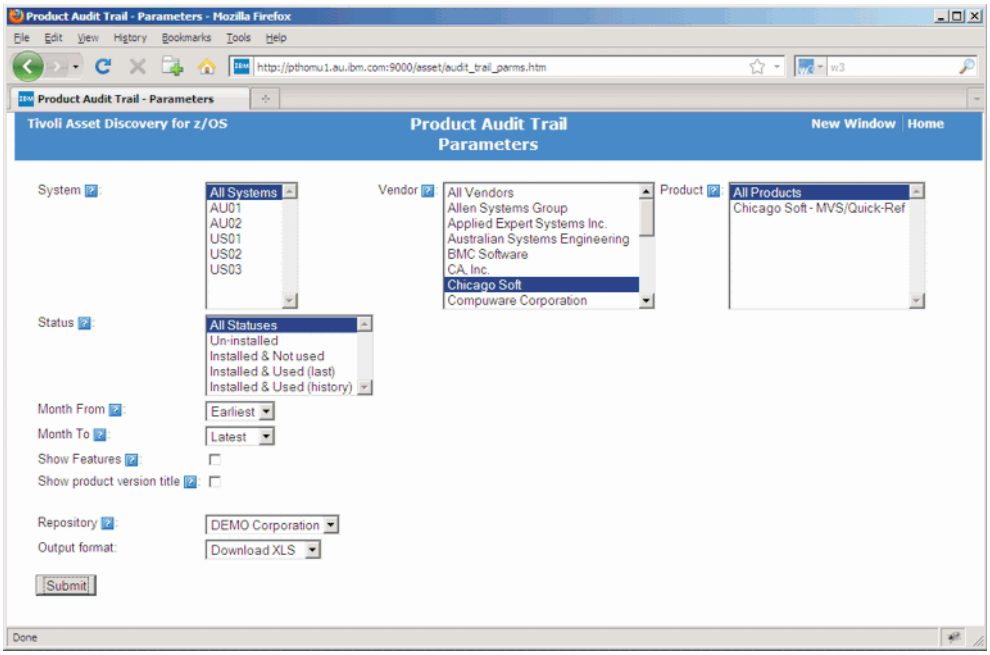
Column	Hyperlink
Discovered Installed	Drill down to **Product by System** report to see more information about products that have been used on Systems which have run on the Machine, or Logical Partition (LPAR).
Systems Installed	Drill down to **Product by System** report to see which systems the product has been discovered on.
Last Used	Drill down to **Product use Trend** report to see product usage trends and details.
Systems Used	Drill down to **Product by System** report
Machines Used	Drill down to **Product Use by Machine** report to see which machines the product has been used on.

Product Audit Trail

Description: Product version audit trail

Batch name: /asset/audit_trail

Sample parameters:



The **Product Audit Trail** report contains a lot of information and is often better displayed as a spreadsheet in which you can filter and sort the data as wanted.

Sample output:

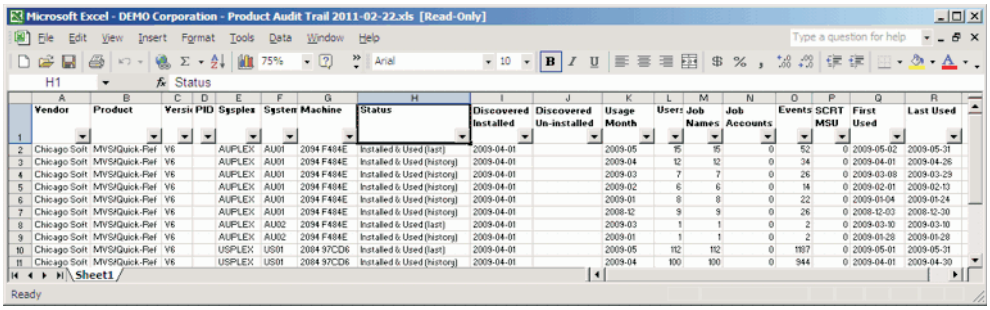


Table 28. Product Audit Trail hyperlinks

Column	Hyperlink
Users	Drill down to see which user IDs have used the product.
Job Names	Drill down to see which job names have used the product.

Table 28. Product Audit Trail hyperlinks (continued)

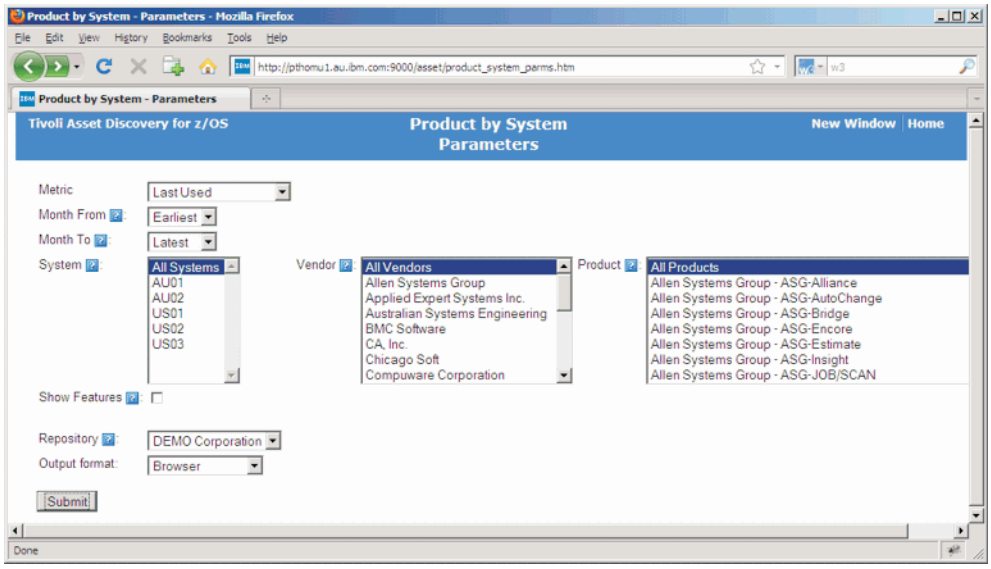
Column	Hyperlink
Job Accounts	Drill down to see which job account codes have used the product.

Product by System

Description: Cross reference of product versions by System

Batch name: /asset/product_system

Sample parameters:



Sample output:

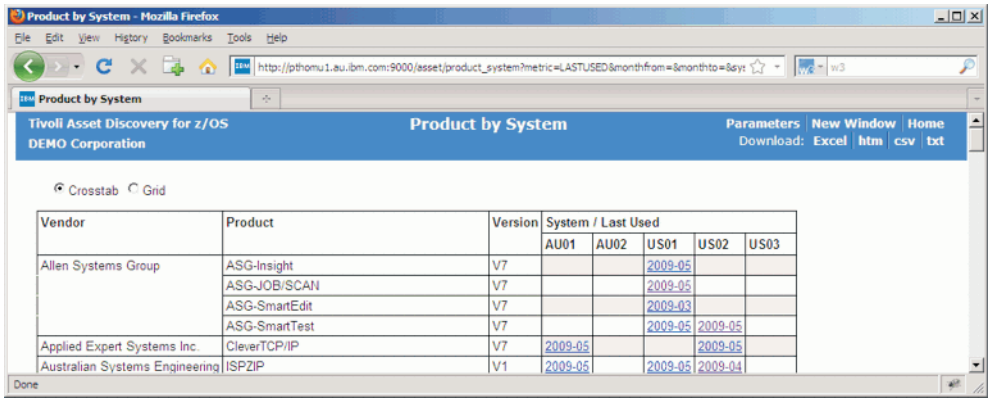


Table 29. Product by System Inventory hyperlinks

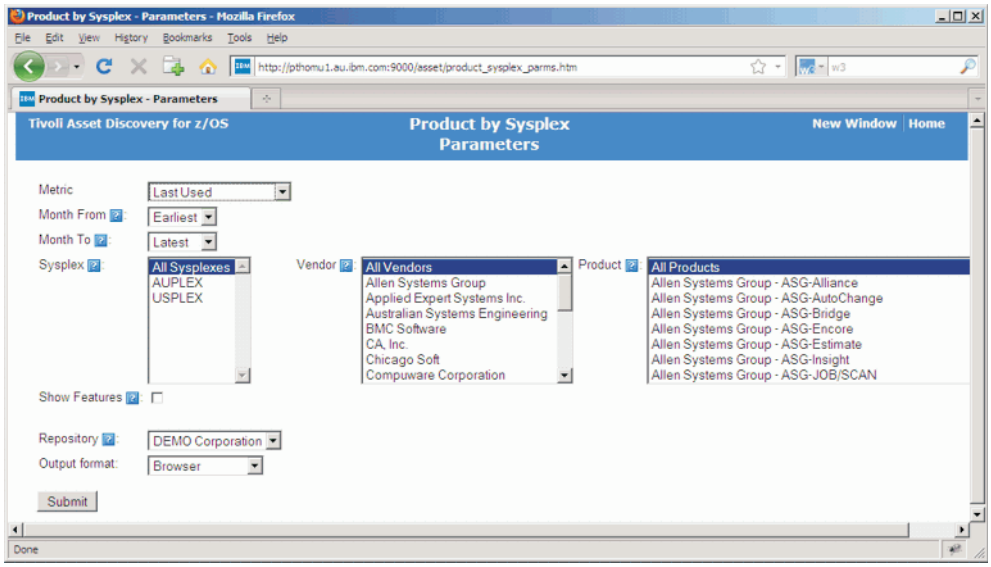
Column	Hyperlink
Last Used First Used Module Events User ID Count Job Name Count Job Account Count SCRT MSU	Drill down to **Product Use Trend** report.
Discovered Installed	Drill down to **Product Libraries report**.

Product by Sysplex

Description: Cross Reference of product versions by Sysplex

Batch name: /asset/product_sysplex

Sample parameters:



Sample output:

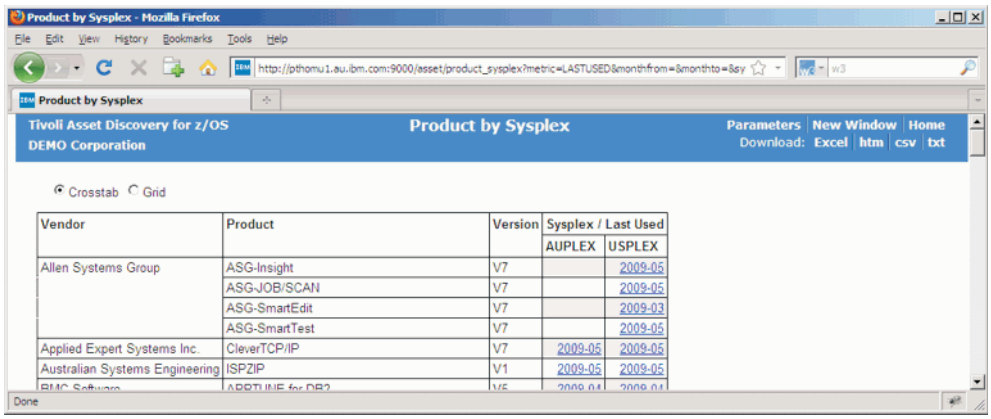


Table 30. Product by Sysples hyperlinks

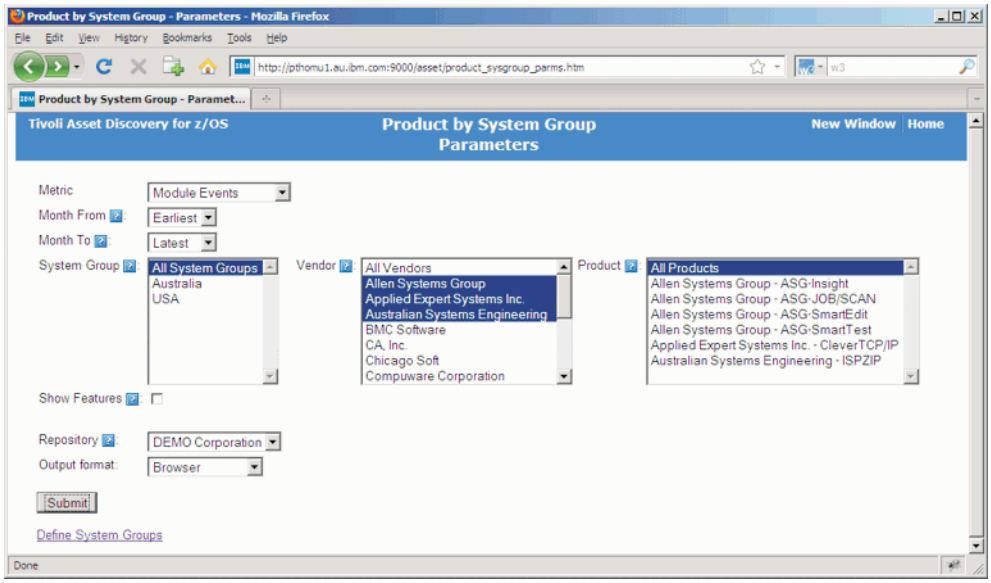
Column	Hyperlink
Last Used First Used Module Events User ID Count Job Name Count Job Account Count SCRT MSU	Drill down to **Product Use Trend** report.
Discovered Installed	Drill down to **Product Libraries report**.

Product by System Group

Description: Cross reference of product versions by System Group

Batch name: /asset/product_sysgroup

Sample parameters:



Sample output:

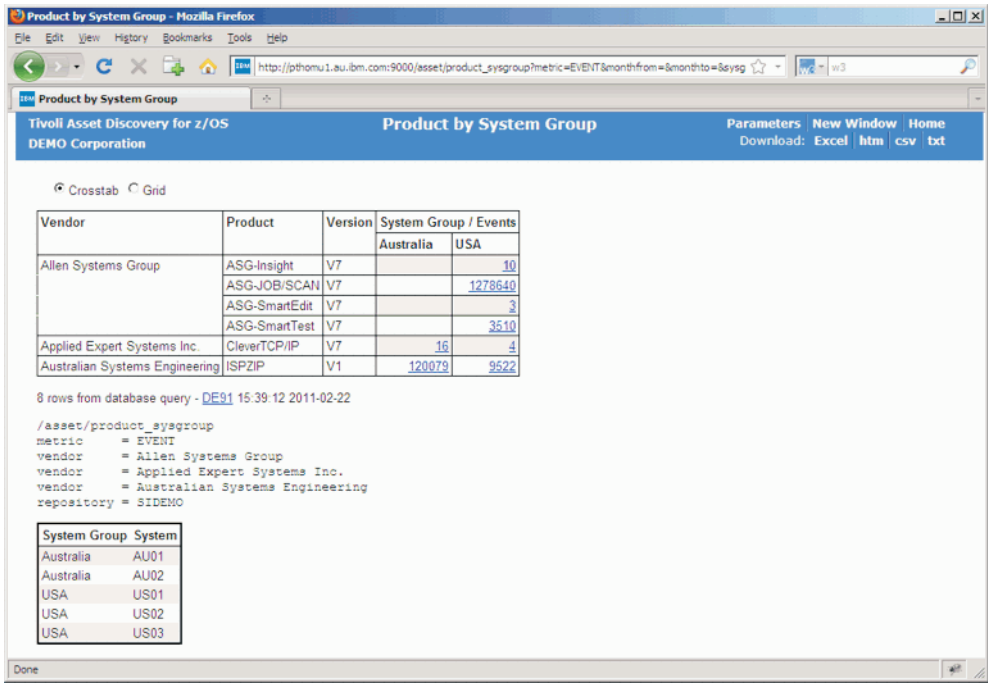


Table 31. Product by System Group hyperlinks

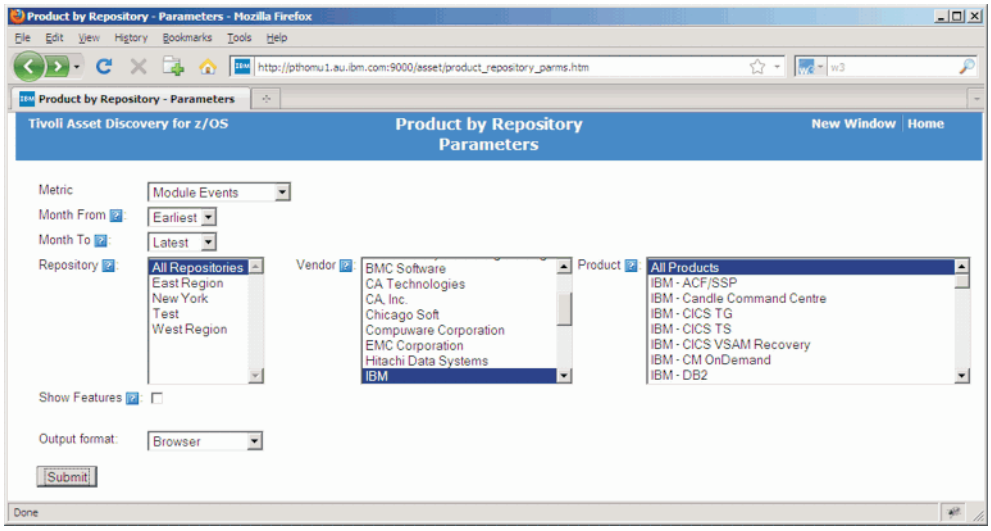
Column	Hyperlink
Last Used First Used Module Events User ID Count Job Name Count Job Account Count SCRT MSU	Drill down to **Product Use Trend** report.
Discovered Installed	Drill down to **Product Libraries report**.

Product by Repository

Description: Cross reference of product versions by Repository

Batch name: /asset/product_repository

Sample parameters:



Sample output:

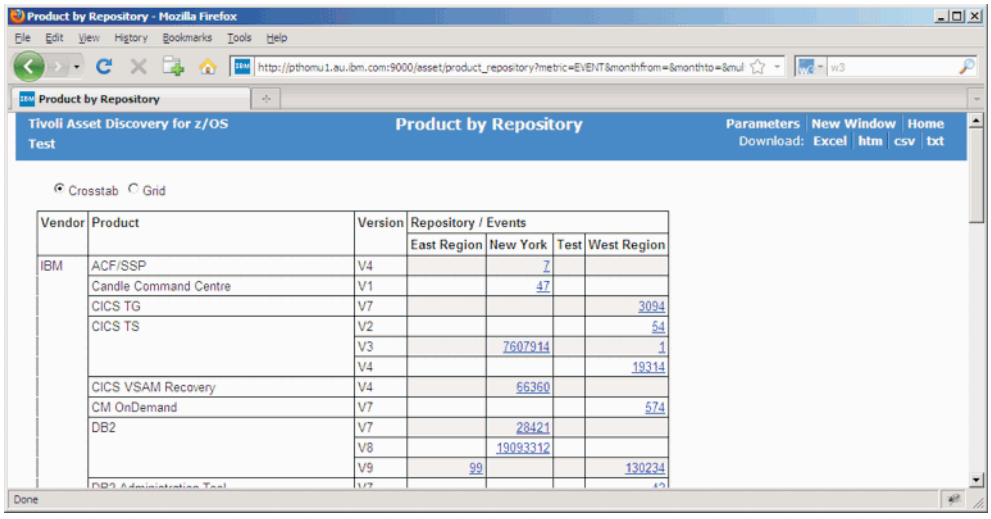


Table 32. Product by Repository hyperlinks

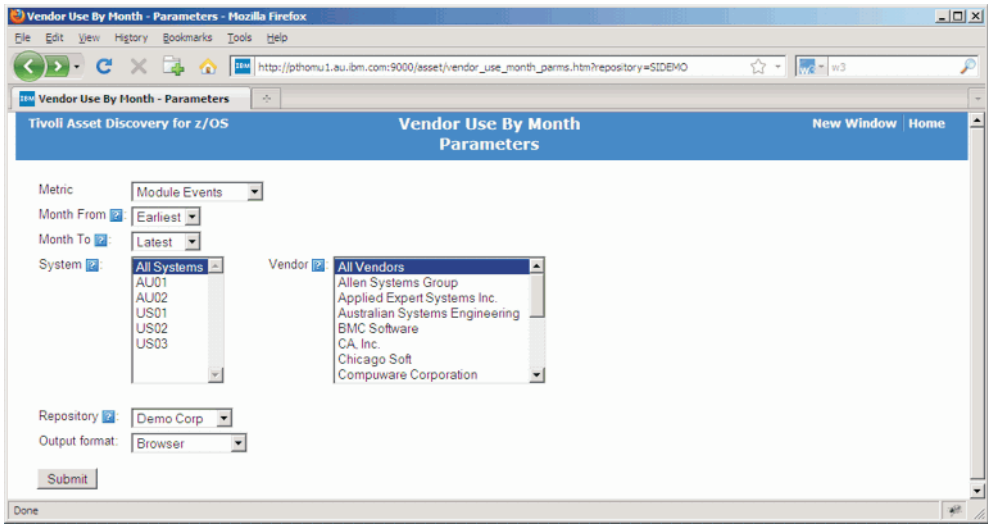
Column	Hyperlink
Last Used First Used Module Events User ID Count Job Name Count Job Account Count SCRT MSU	Drill down to **Product Use Trend** report.
Discovered Installed	Drill down to **Product Libraries report**.

Vendor Use by Month

Description: Cross reference of vendor products used by Month by System

Batch name /asset/vendor_use_month

Sample parameters:



Sample output:

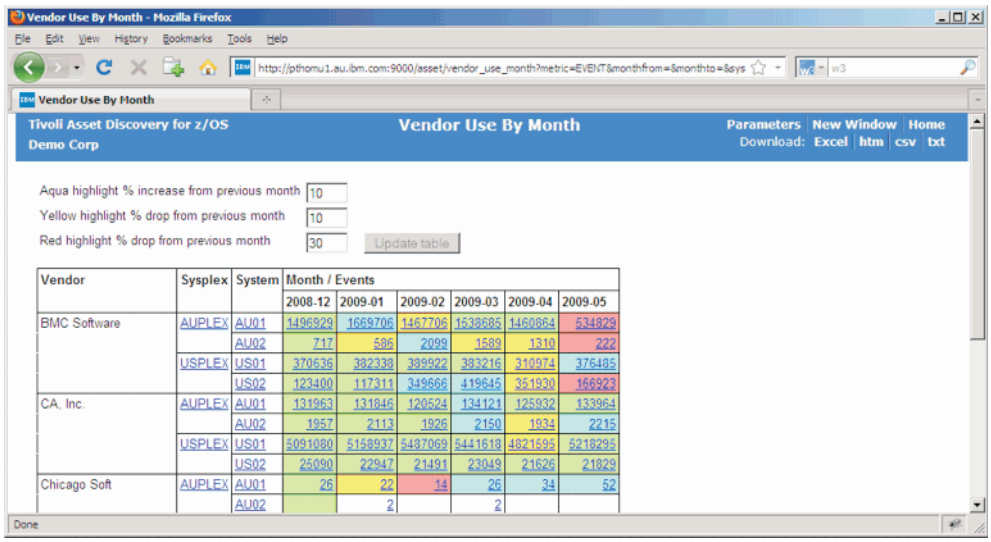


Table 33. Vendor Use by Month hyperlinks

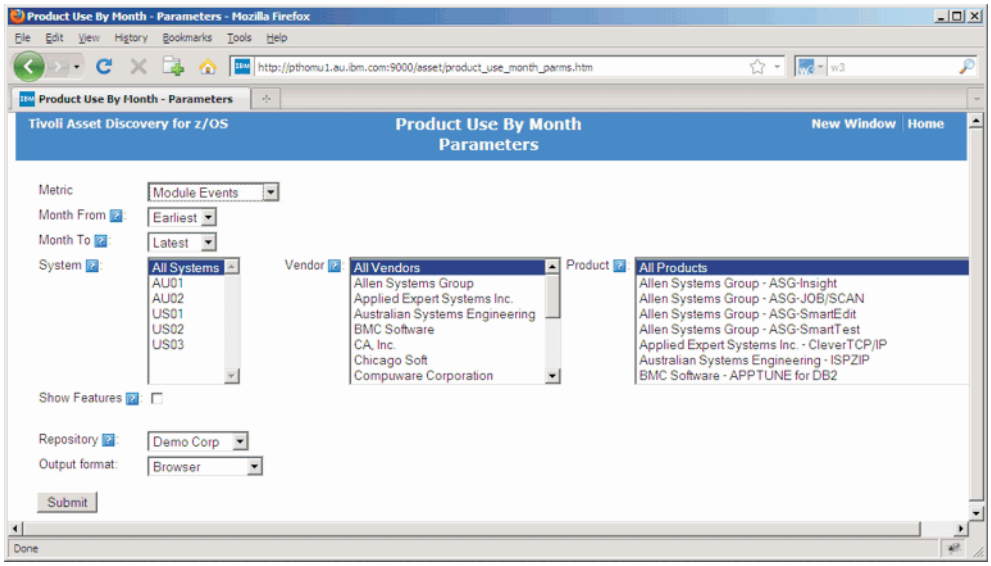
Column	Hyperlink
Sysplex	Drill down to **Product Use Trend** report.
System	Drill down to **Product Use Trend** report.
Events Users SCRT MSU	Drill down to see which user IDs have used the product.
Job Names	Drill down to see which job names have used the product.
Job Accounts	Drill down to see which job account codes have used the product.

Product Use by Month

Description: Cross reference of product versions used per Month by System

Batch name: /asset/product_use_month

Sample parameters:



Sample output:

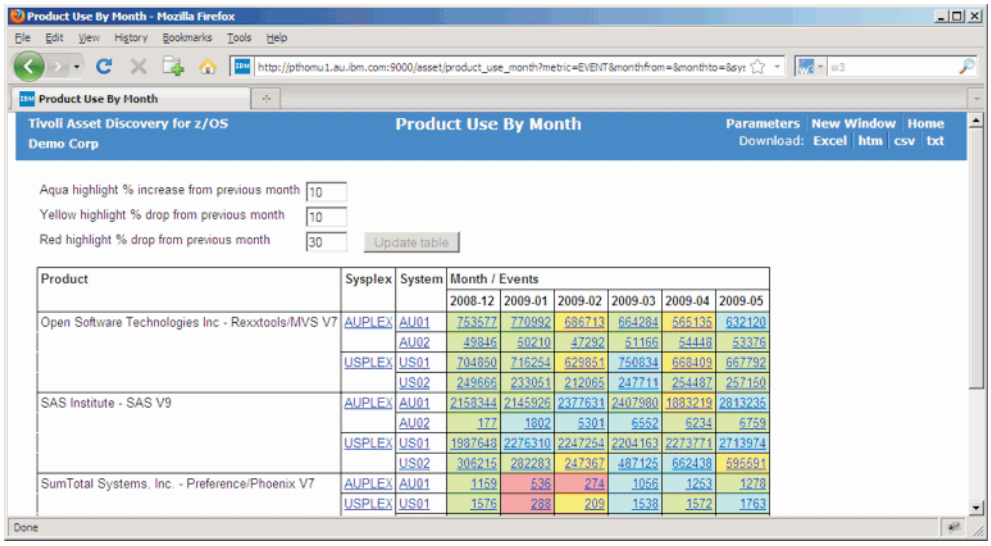


Table 34. Product Use by Month hyperlinks

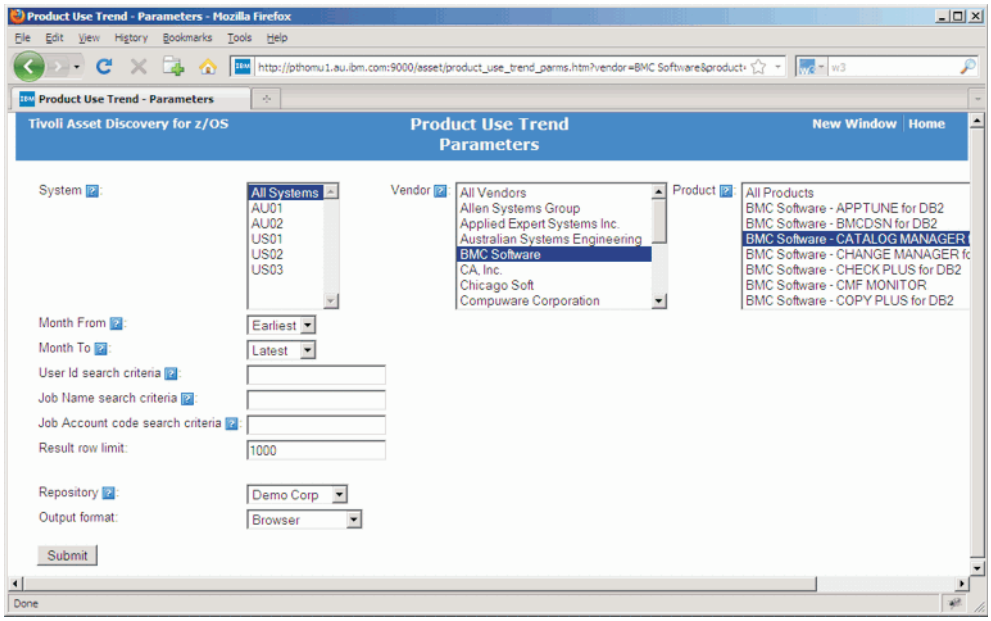
Column	Hyperlink
Sysplex	Drill down to **Product Use Trend** report.
System	Drill down to **Product Use Trend** report.
Events Users SCRT MSU	Drill down to see which user IDs have used the product.
Job Names	Drill down to see which job names have used the product.
Job Accounts	Drill down to see which job account codes have used the product.

Product Use Trend

Description: Product version usage trend chart

Batch name: /asset/product_use_trend

Sample parameters:



Mouse over the chart to see the values.

Sample output:

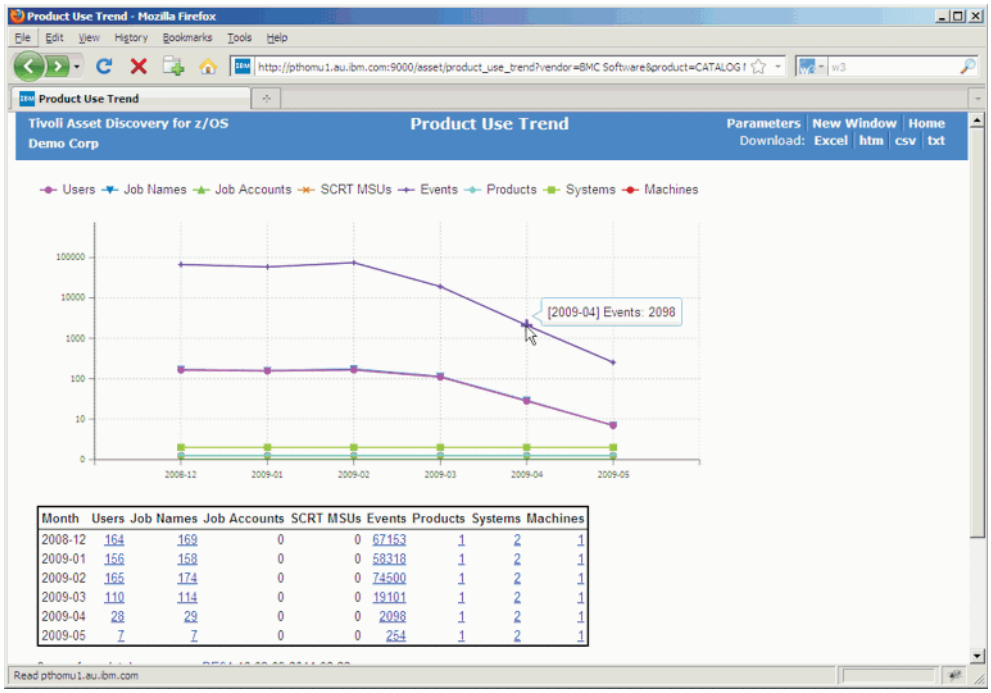


Table 35. Product Use Trend hyperlinks

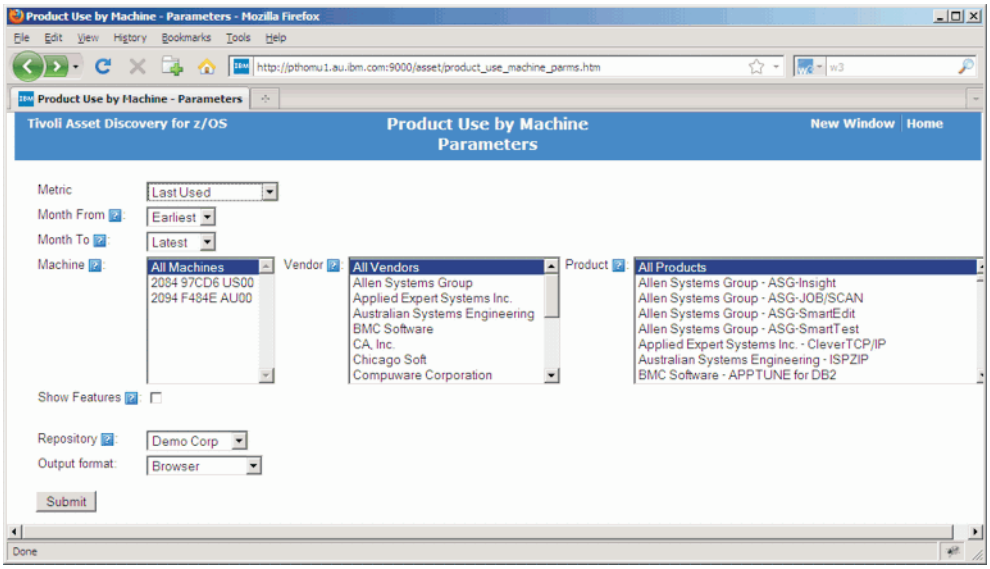
Column	Hyperlink
Users	Drill down to see which user IDs have used the product.
Job Names	Drill down to see which job names have used the product.
Job Accounts	Drill down to see which job account codes have used the product.
Events	Drill down to **Product Library Usage** report.
	Drill down to **Product Inventory** report.
	Drill down to **Product by System** report.
	Drill down to **Product Use by Machine** report.

Product Use by Machine

Description: Cross reference of product versions used by machine

Batch name: /asset/product_use_machine

Sample parameters:



Sample output:

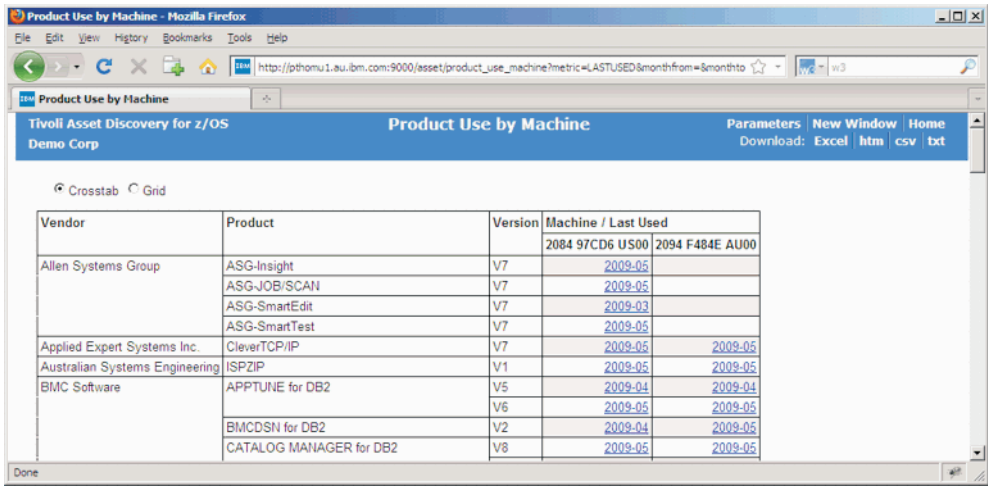


Table 36. Product Use by Machine hyperlinks

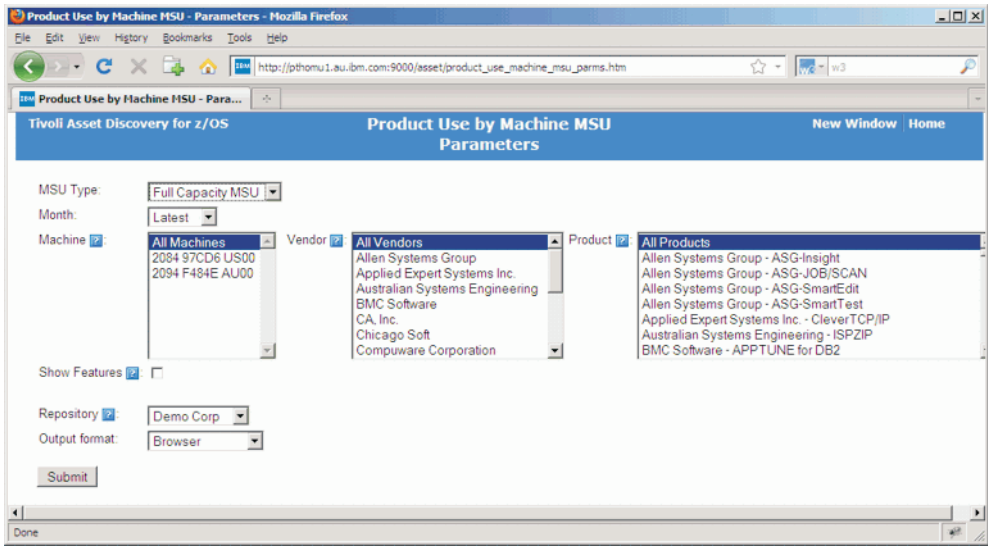
Column	Hyperlink
Last Used First Used Module Events User ID Count Job Name Count Job Account Count SCRT MSU	Drill down to **Product Use Trend** report.

Product Use by Machine MSU

Description: Cross reference of product versions by Machine MSU capacity

Batch name: /asset/product_use_machine_msu

Sample parameters:



Sample output:

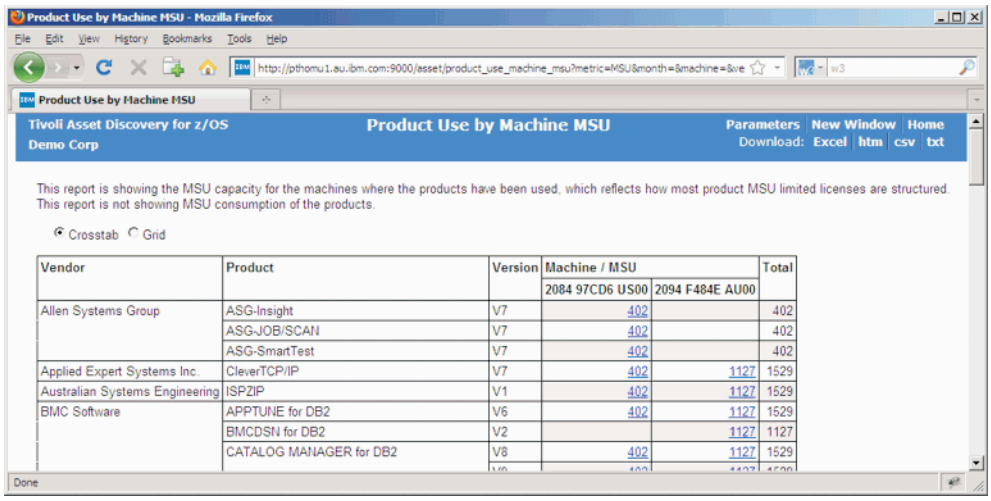


Table 37. Product Use by Machine MSU hyperlinks

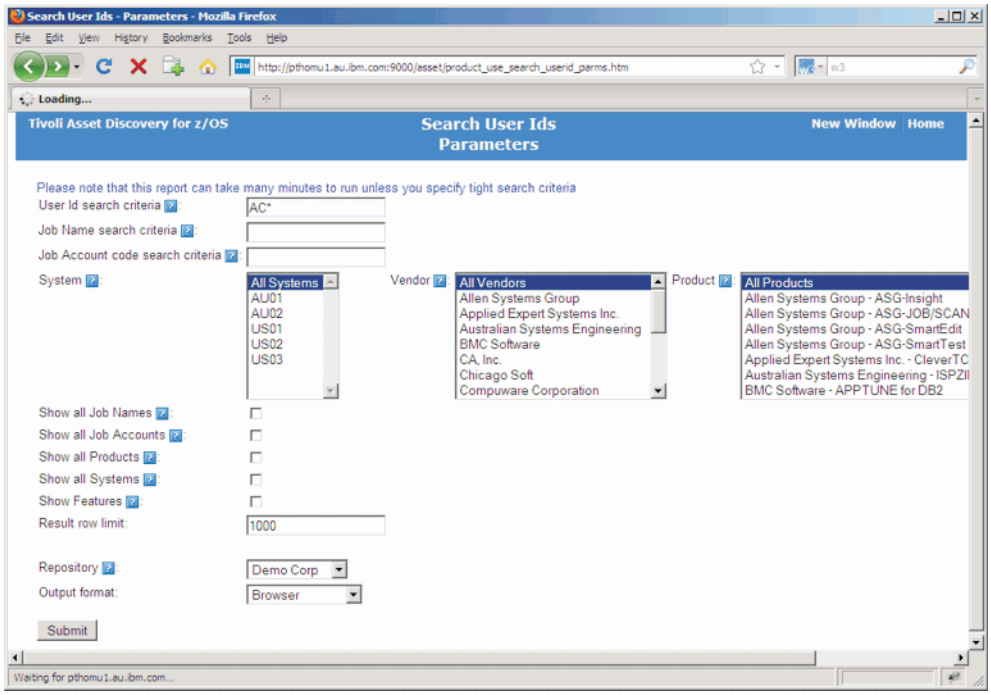
Column	Hyperlink
MSU	Drill down to **Product Use Trend** report.

Search User Ids

Description: Search product version usage details for user IDs

Batch name: /asset/product_use_search_userid

Sample parameters:



Sample output:

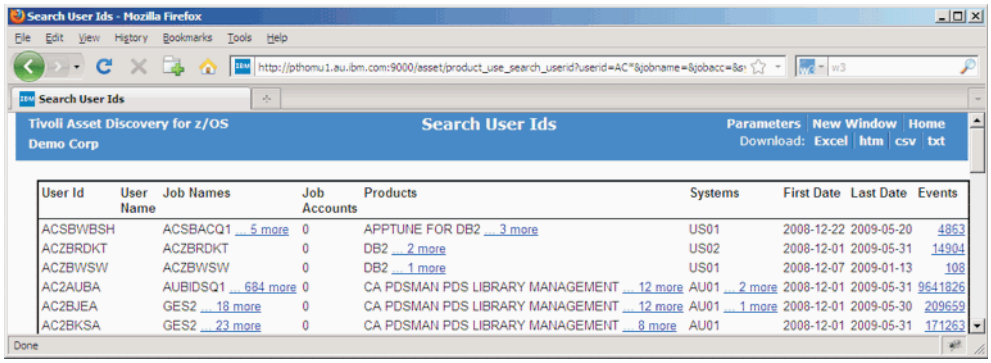


Table 38. Sear User Ids hyperlinks

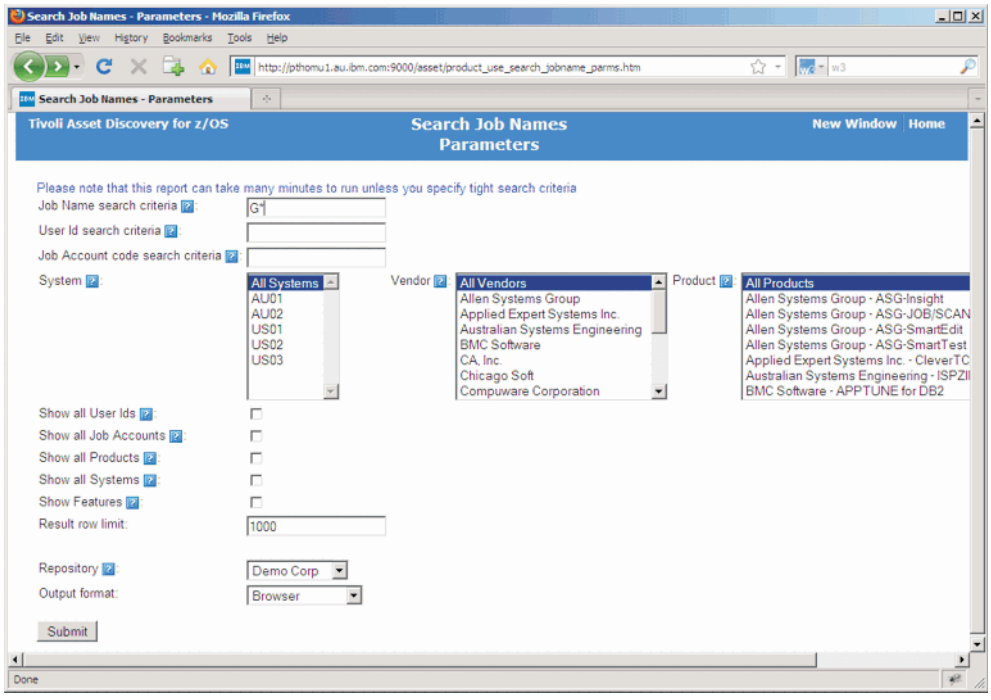
Column	Hyperlink
Job Names	Drill down to see which job names have used the product.
Job Accounts	Drill down to see which job account codes have used the product.
Products	Drill down to Search user IDs to show all Products.
Systems	Drill down to Search user IDs to show all Systems.
Events	Drill down to see module usage details.

Search Job Names

Description: Search product version usage details for job names

Batch name: /asset/product_use_search_jobname

Sample parameters:



Sample output:

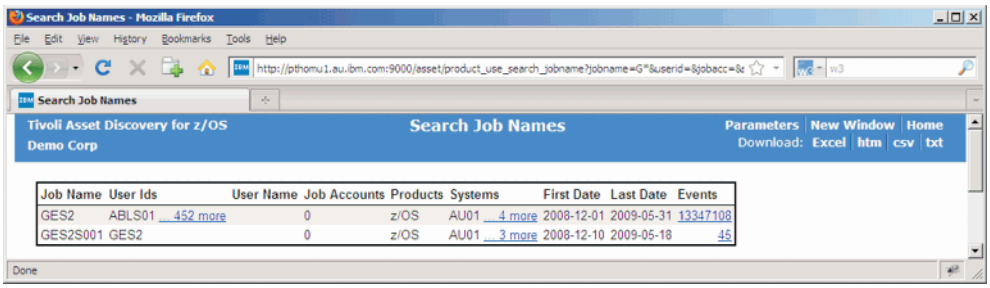


Table 39. Search Job Names hyperlinks

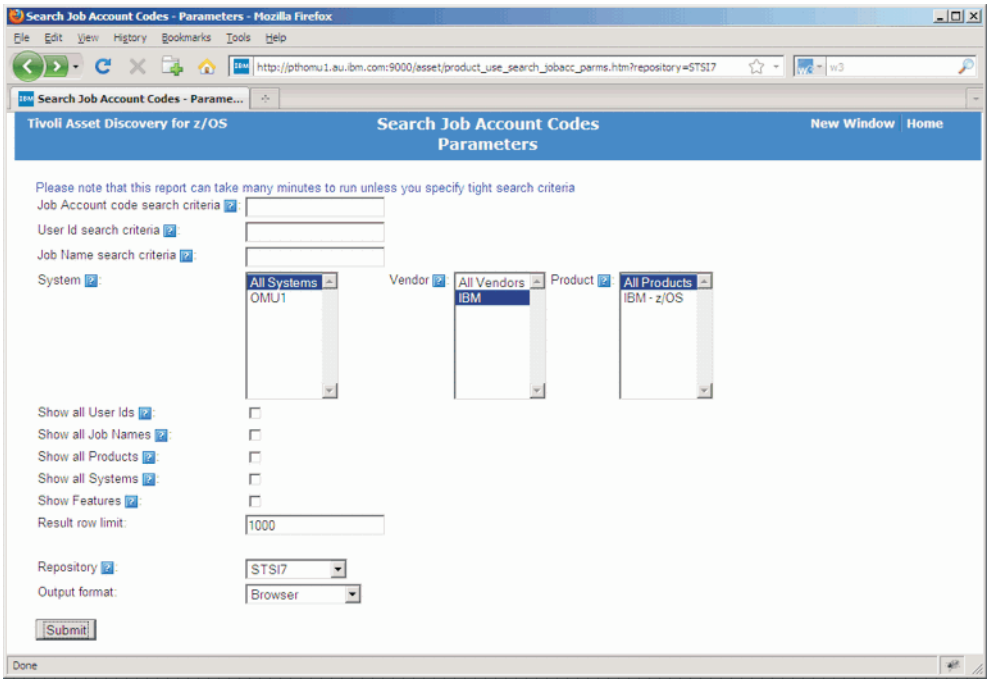
Column	Hyperlink
User IDs	Drill down to Search Users.
Job Accounts	Drill down to Search Job Accounts.
Products	Drill down to Search Job Names to show all Products.
Systems	Drill down to Search Job Names to show all Systems.
Events	Drill down to see module usage details.

Search Job Account Codes

Description: Search product version usage details for job account codes

Batch name: /asset/product_use_search_jobacc

Sample parameters:



Sample output:

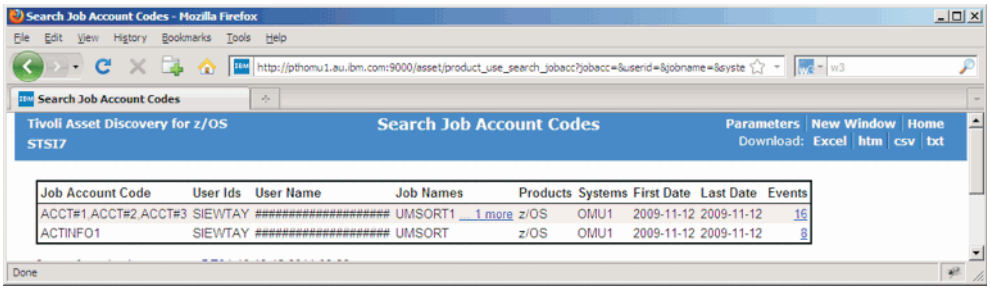


Table 40. Search Job Account Codes hyperlinks

Column	Hyperlink
User IDs	Drill down to Search Users.
Job Accounts	Drill down to Search Job Accounts.
Products	Drill down to Search Job Names to show all Products.
Systems	Drill down to Search Job Names to show all Systems.
Events	Drill down to see module usage details.

Analyzer Discovery reports

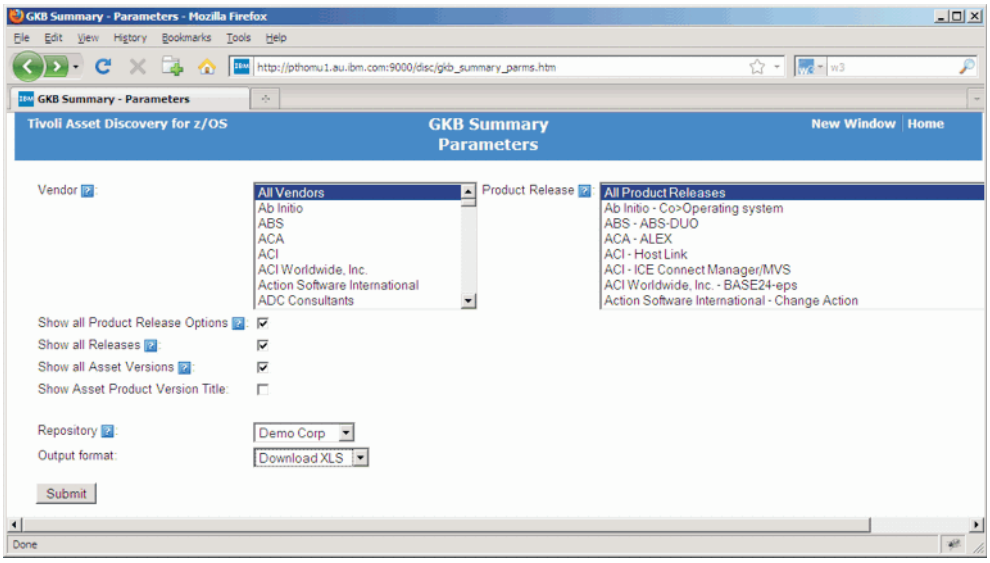
This section contains a list of the discovery reports. For each report, two sample screens are provided. The first screen displays sample parameters, followed by sample output. Directly following these screens is a table which describes the function of each hyperlink in the output section of the report.

GKB Summary

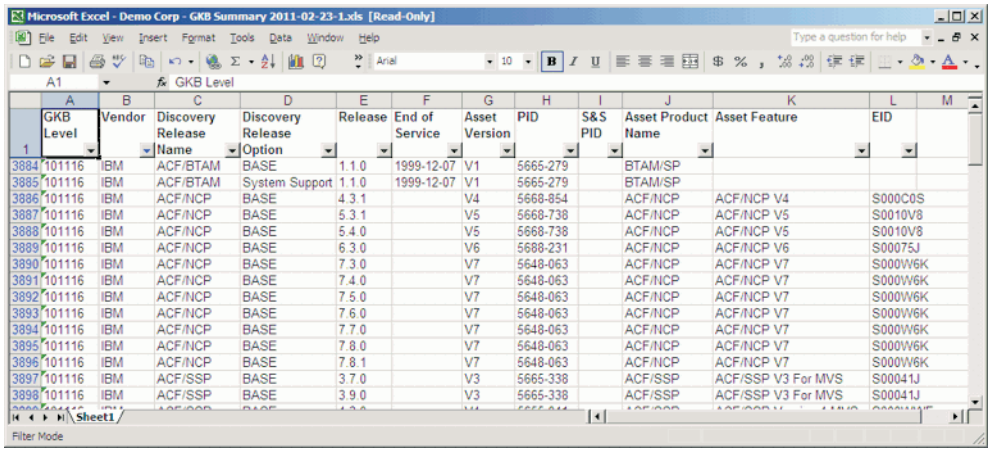
Description: Summary of products in the Global Knowledge Base catalog

Batch name: /disc/gkb_summary

Sample parameters:



Sample output:



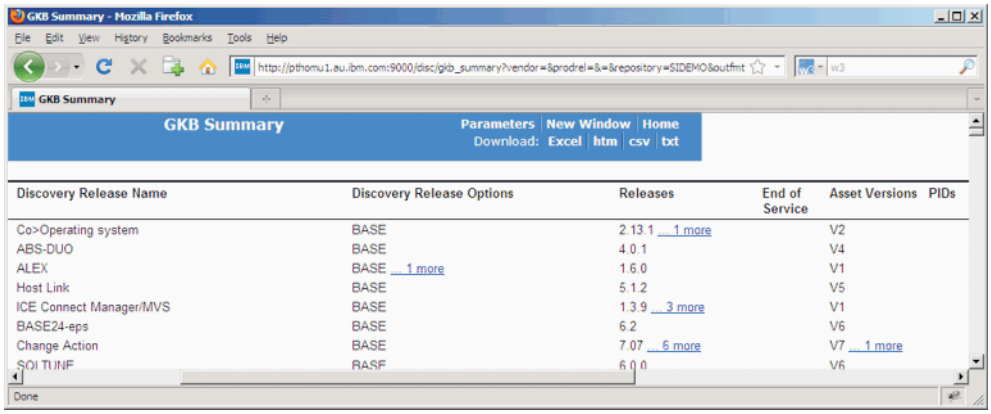


Table 41. GKB Summary hyperlinks

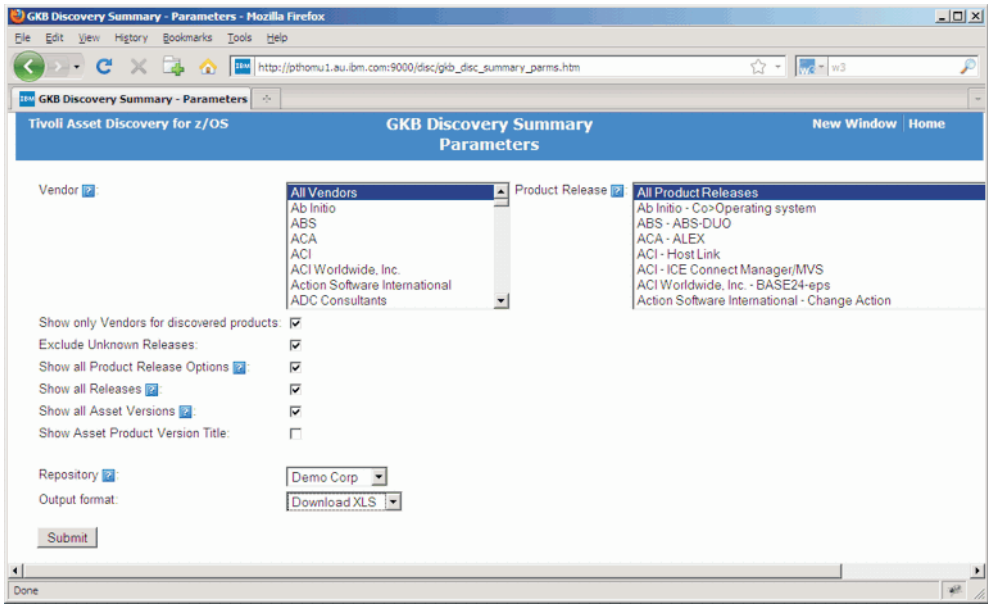
Column	Hyperlink
Discovery Release Options	Drill down to GKB Summary to see more Product Release Options
Releases	Drill down to GKB Summary to see more Releases
Asset Versions	Drill down to GKB Summary to see more Asset Versions

GKB Discovery Summary

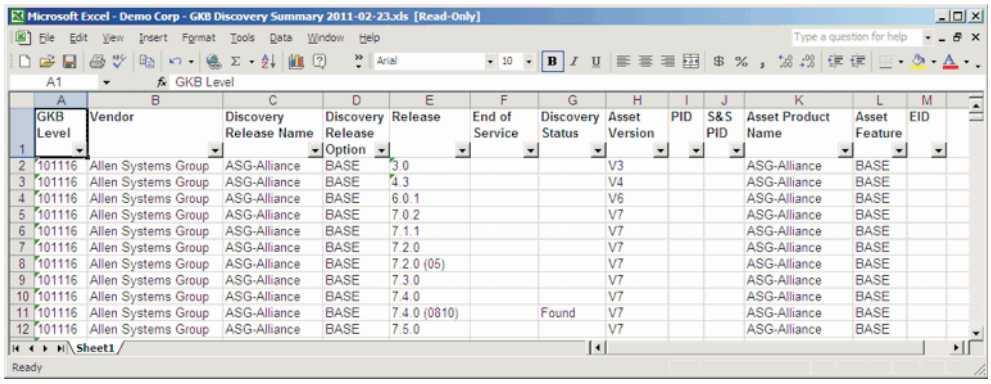
Description: Summary of products in the Global Knowledge Base catalog showing which products have been discovered

Batch name: /disc/gkb_disc_summary

Sample parameters:



Sample output:



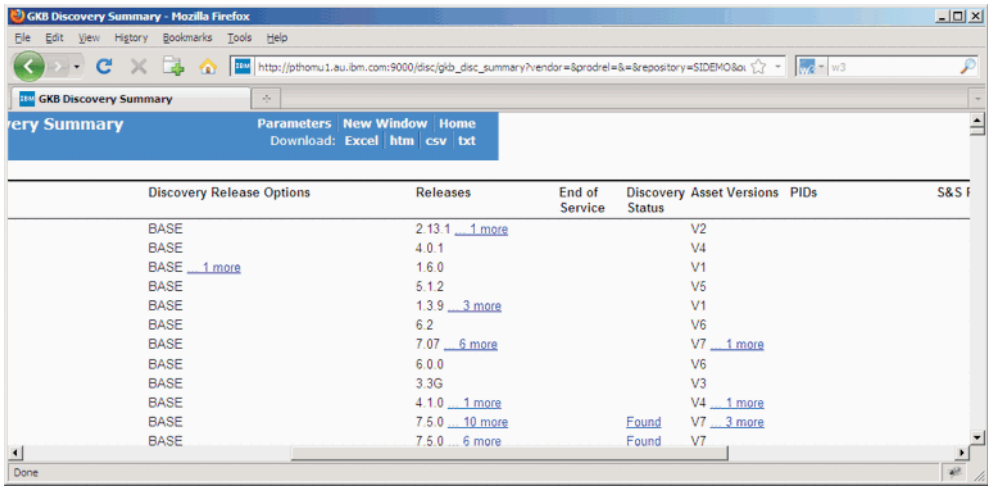


Table 42. GKB Discovery Summary hyperlinks

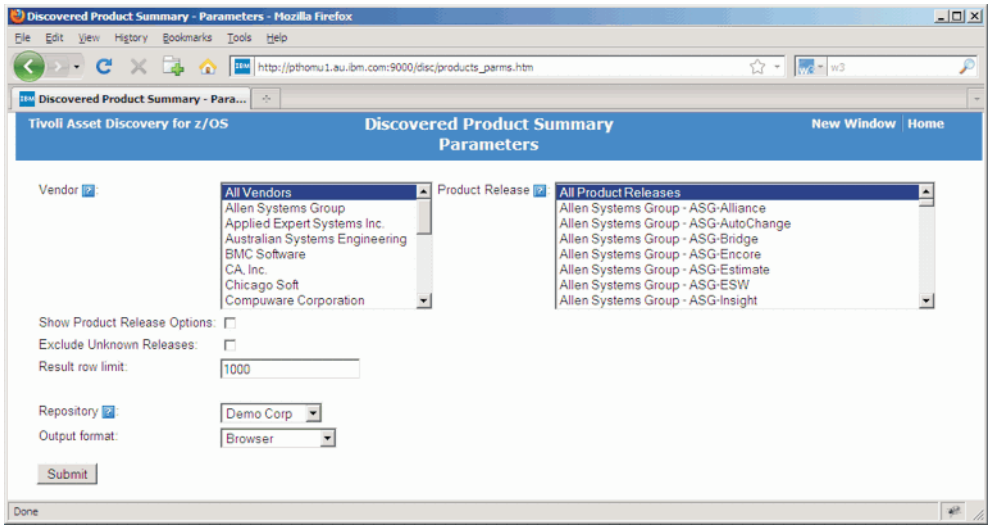
Column	Hyperlink
Discovery Release Options	Drill down to GKB Summary to see more Product Release Options
Releases	Drill down to GKB Summary to see more Releases
Discovery Status	Drill down to Product Libraries
Asset Versions	Drill down to GKB Summary to see more Asset Versions

Discovered Product Summary

Description: Summary of discovered products

Batch name: /disc/products

Sample parameters:



Sample output:

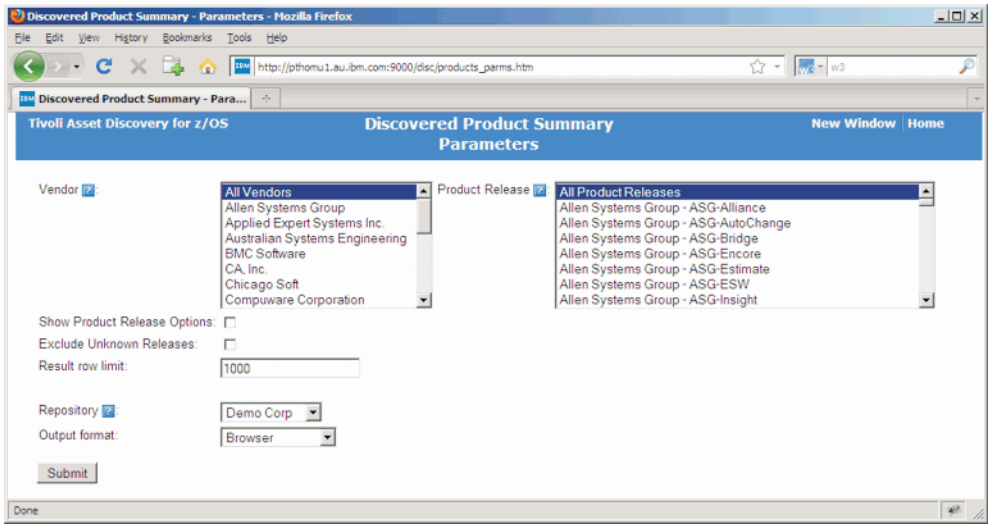


Table 43. Discovered Product Summary hyperlinks

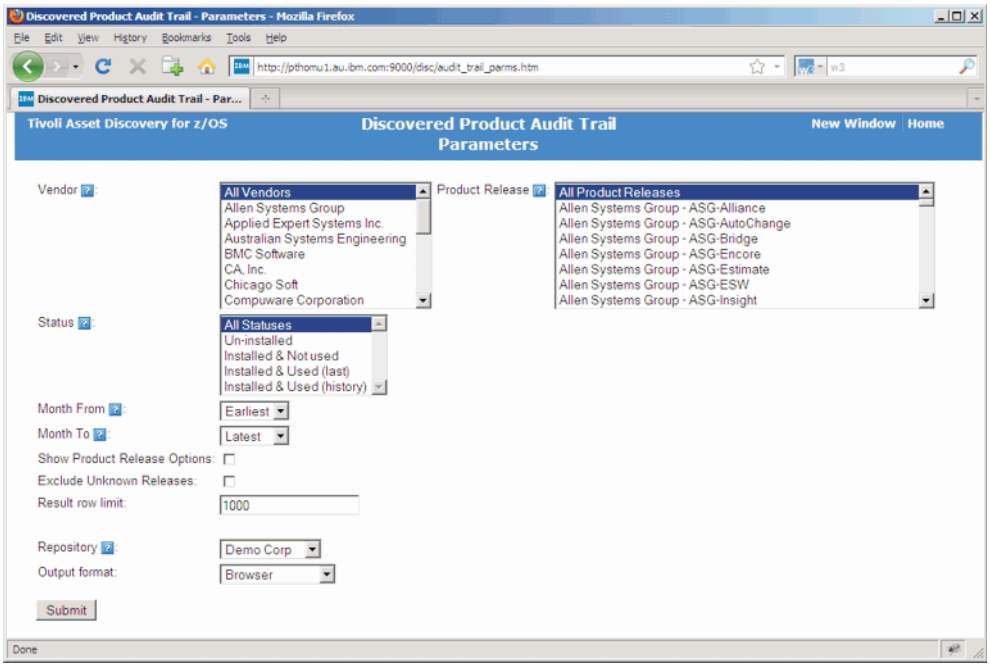
Column	Hyperlink
Asset Version	Drill down to Product Inventory
Libraries	Drill down to Product Libraries
Last Used	Drill down to Product Library Usage

Discovered Product Audit Trail

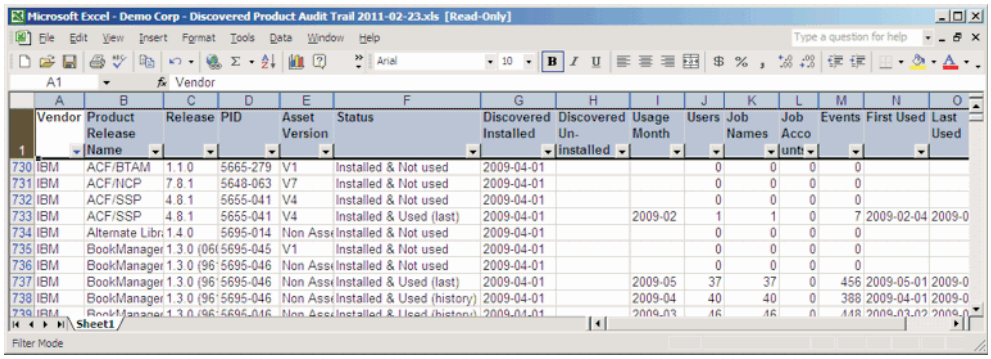
Description: Audit trail of discovered products

Batch name: /disc/audit_trail

Sample parameters:



Sample output:



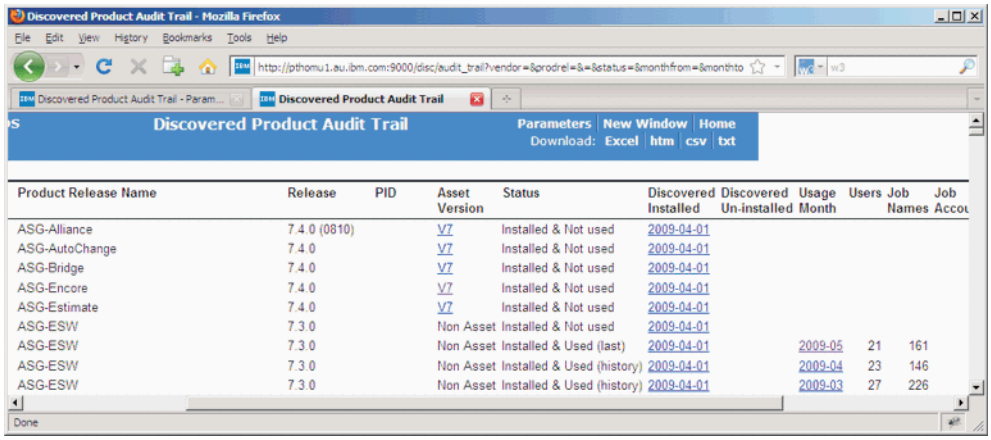


Table 44. Discovered Product Audit Trail hyperlinks

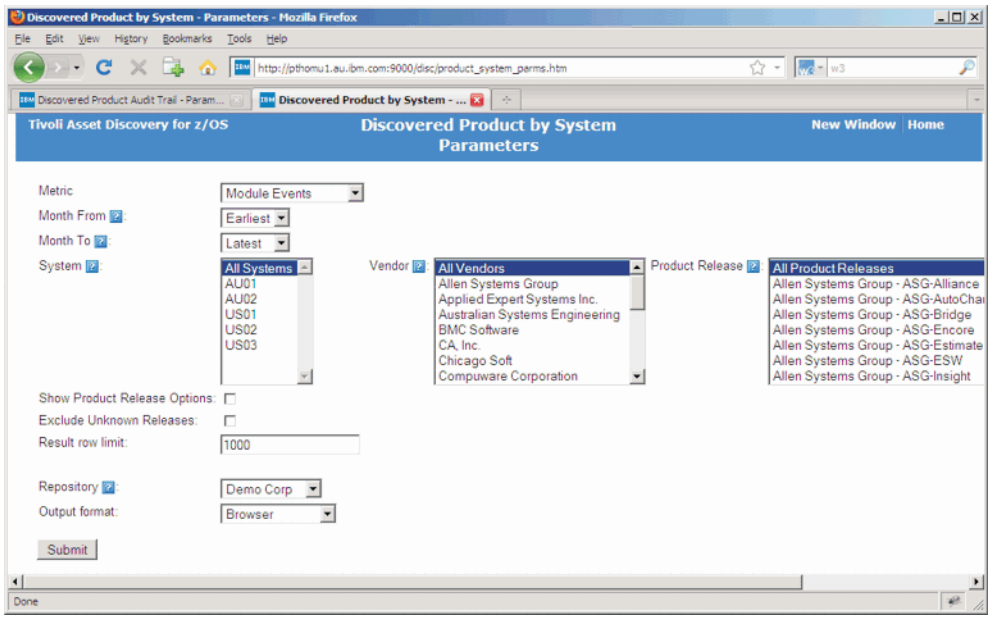
Column	Hyperlink
Asset Version	Drill down to Product Inventory
Discovered Installed	Drill down to Product Libraries
Usage Month	Drill down to Product Library Usage

Discovered Product by System

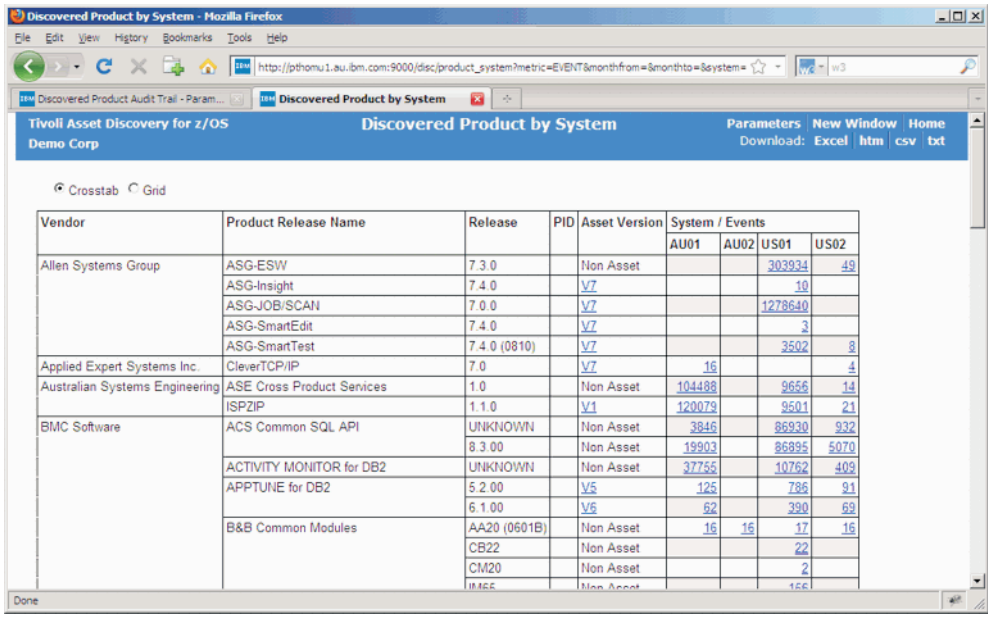
Description: Cross-reference of discovered products by System

Batch name: /disc/product_system

Sample parameters:



Sample output:



Vendor	Product Release Name	Release	PID	Asset Version	System / Events			
					AU01	AU02	US01	US02
Allen Systems Group	ASG-ESW	7.3.0		Non Asset			303934	49
	ASG-Insight	7.4.0		V7			10	
	ASG-JOB/SCAN	7.0.0		V7			1278640	
	ASG-SmartEdit	7.4.0		V7			3	
	ASG-SmartTest	7.4.0 (0810)		V7			3502	8
Applied Expert Systems Inc.	CleverTCP/IP	7.0		V7	16			4
Australian Systems Engineering	ASE Cross Product Services	1.0		Non Asset	104488		9656	14
	ISPZIP	1.1.0		V1	120079		9501	21
BMC Software	ACS Common SQL API	UNKNOWN		Non Asset	3846		86930	932
		8.3.00		Non Asset	19903		86895	5070
	ACTIVITY MONITOR for DB2	UNKNOWN		Non Asset	37755		10762	409
	APPTUNE for DB2	5.2.00		V5	125		786	91
		6.1.00		V6	62		390	69
	B&B Common Modules	AA20 (0601B)		Non Asset	16	16	17	16
		CB22		Non Asset			22	
		CM20		Non Asset			2	

Table 45. Discovered Product by System hyperlinks

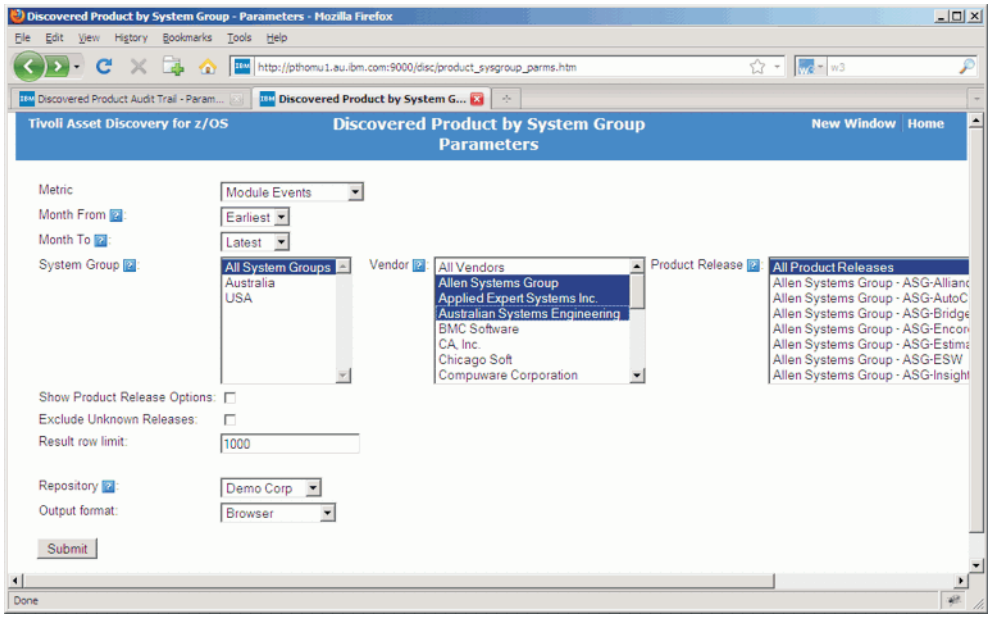
Column	Hyperlink
Asset Version	Drill down to Product Inventory
Last Used Module Events User ID Count Job Name Count Job Account Count	Drill down to Product Library Usage
Discovered Installed	Drill down to Product Libraries

Discovered Product by System Group

Description: Cross-reference of discovered products per System Group

Batch name: /disc/product_sysgroup

Sample parameters:



Sample output:

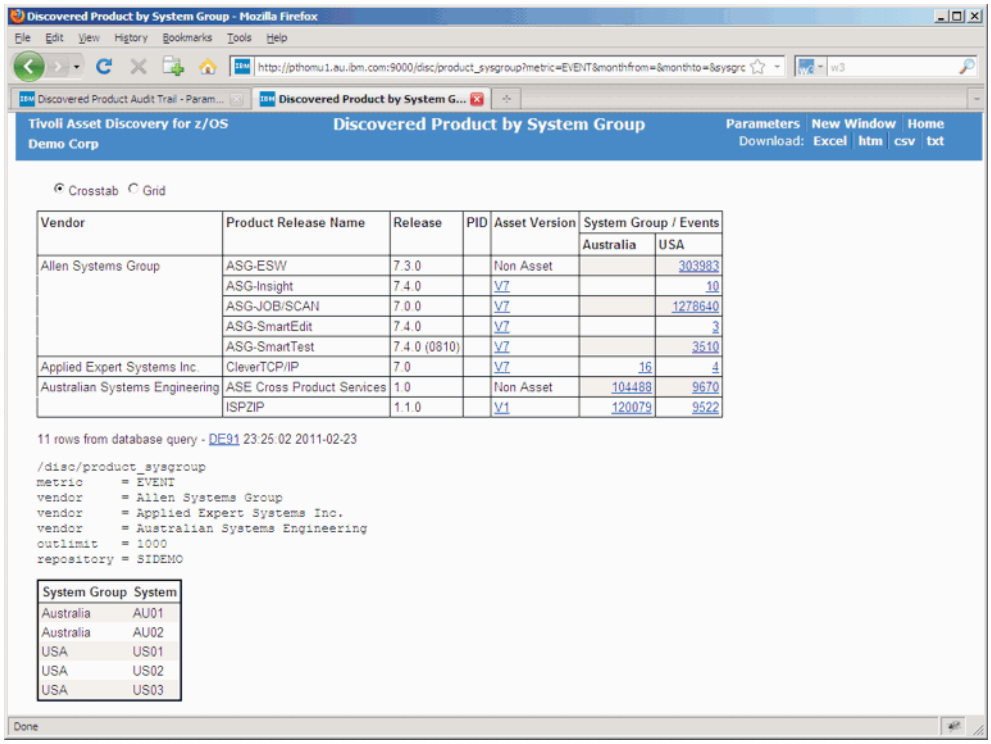


Table 46. Discovered Product by System Group hyperlinks

Column	Hyperlink
Asset Version	Drill down to Product Inventory

Table 46. Discovered Product by System Group hyperlinks (continued)

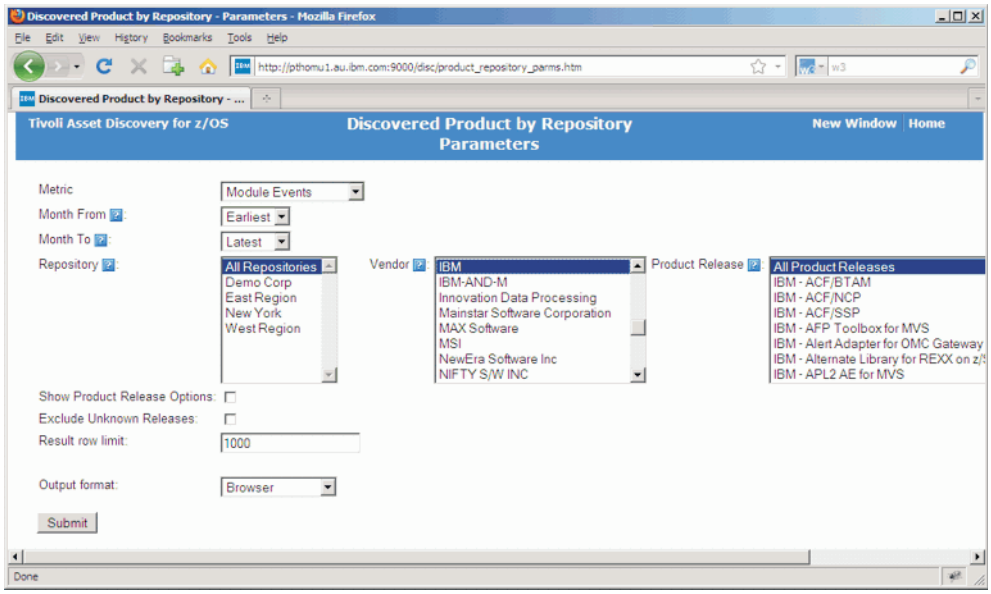
Column	Hyperlink
Last Used Module Events User ID Count Job Name Count Job Account Count	Drill down to Product Library Usage
Discovered Installed	Drill down to Product Libraries

Discovered Product by Repository

Description: Cross-reference of discovered products by Repository

Batch name: /disc/product_repository

Sample parameters:



Sample output:

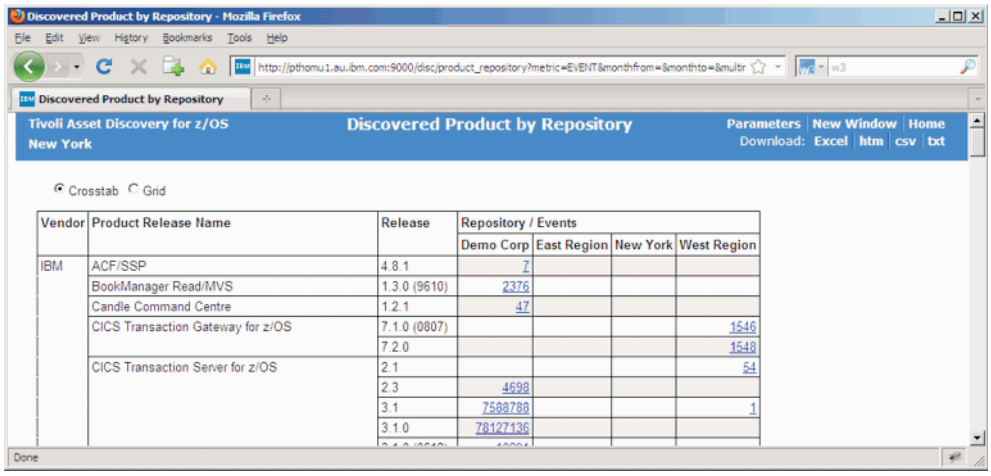


Table 47. Discovered Product by Repository hyperlinks

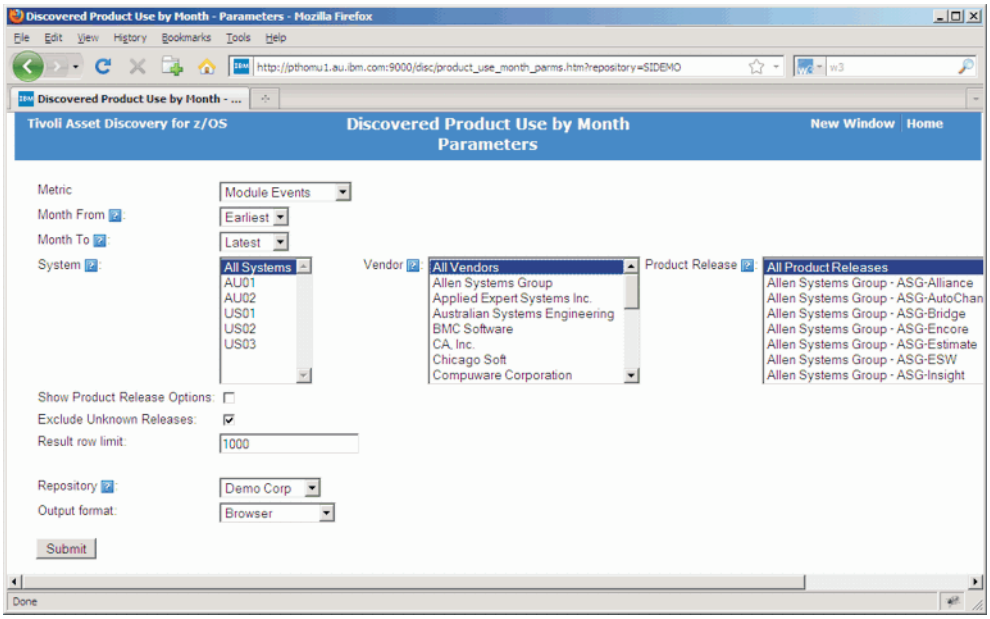
Column	Hyperlink
Last Used Module Events User ID Count Job Name Count Job Account Count	Drill down to Product Library Usage
Discovered Installed	Drill down to Product Libraries

Discovered Product Use by Month

Description: Cross-reference of discovered products used per Month by System

Batch name: /disc/product_use_month

Sample parameters:



Sample output:

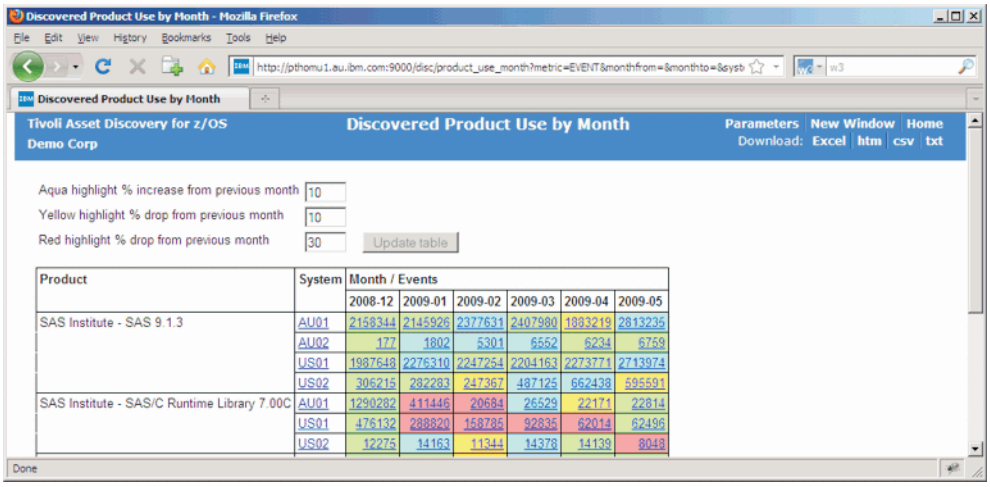


Table 48. Discovered Product Use by Month hyperlinks

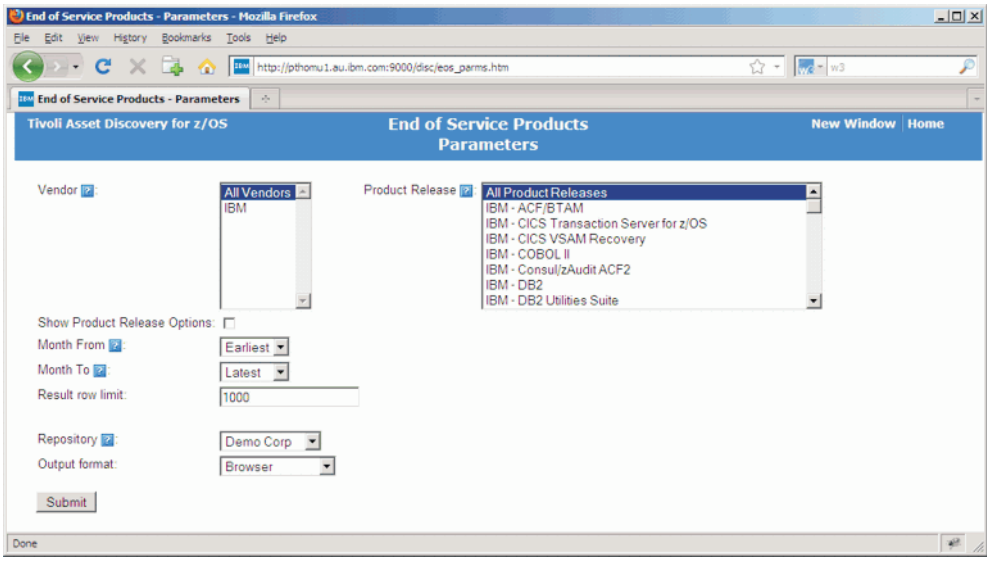
Column	Hyperlink
System Events User ID Count Job Name Count Job Account Count	Drill down to Product Library Usage
Discovered Installed	Drill down to Product Libraries

End of Service Products

Description: Summary of discovered products that have a known End of Service date

Batch name: /disc/eos

Sample parameters:



Sample output:

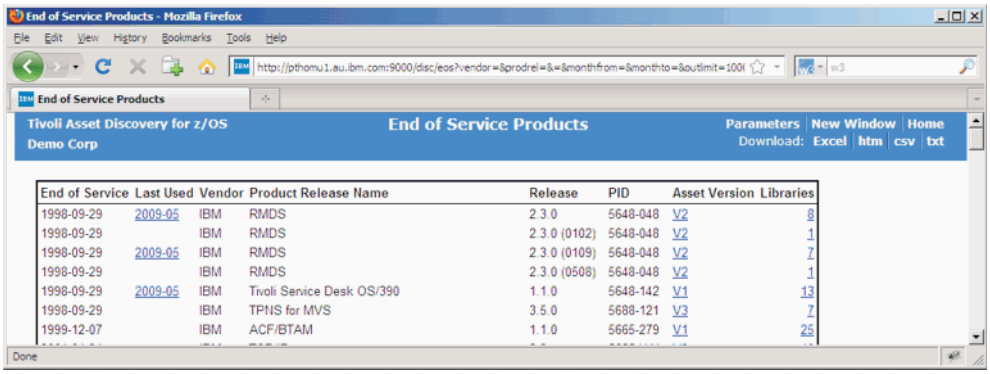


Table 49. Discovered Product Use by Month hyperlinks

Column	Hyperlink
Last Used	Drill down to Product Library Usage

Table 49. Discovered Product Use by Month hyperlinks (continued)

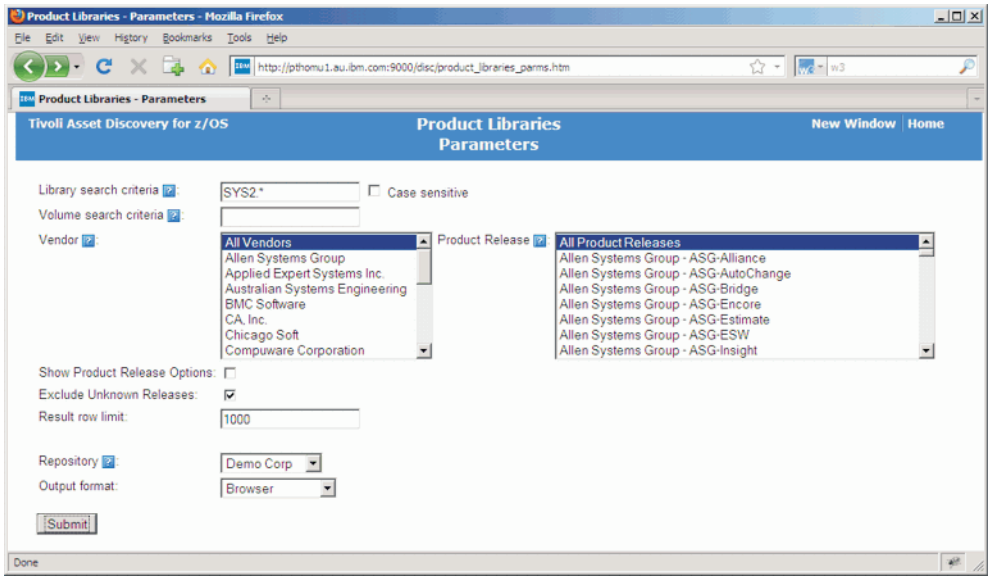
Column	Hyperlink
Asset Version	Drill down to Product Inventory

Product Libraries

Description: Summary of discovered product libraries

Batch name: /disc/product_libraries

Sample parameters:



Sample output:

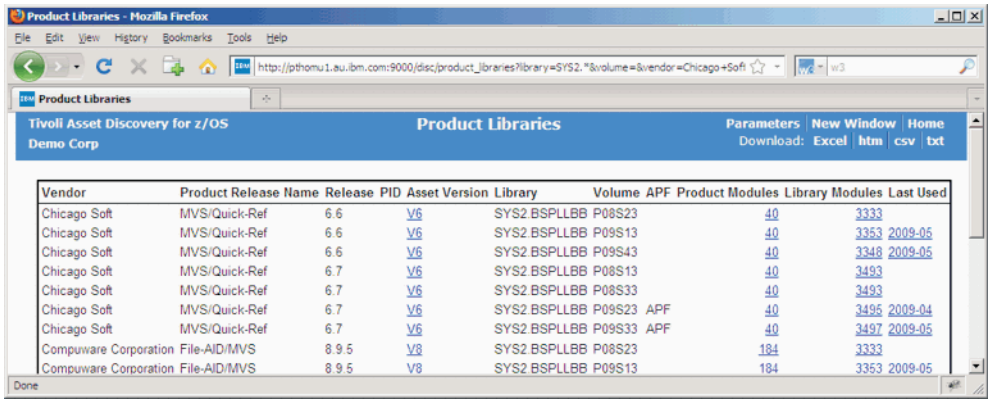


Table 50. Product Libraries hyperlinks

Column	Hyperlink
Asset Version	Drill down to Product Inventory
Product Modules	Drill down to Search Modules to show the Product modules in the library
Library Modules	Drill down to Search Modules to show all modules in the library

Table 50. Product Libraries hyperlinks (continued)

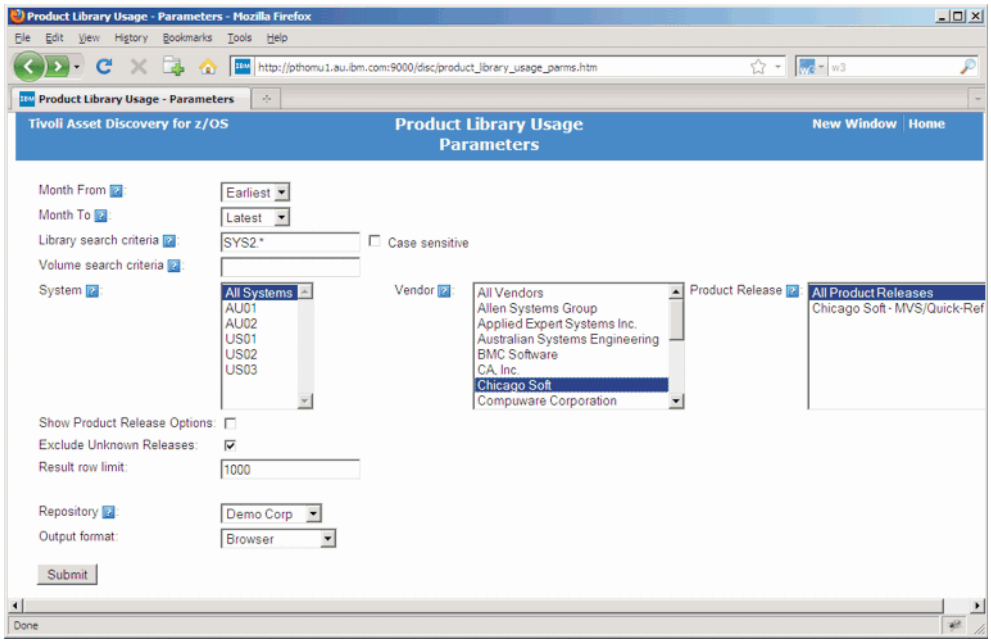
Column	Hyperlink
Last Used	Drill down to Product Library Usage

Product Library Usage

Description: Summary of discovered product library usage

Batch name: /disc/product_library_usage

Sample parameters:



Sample output:

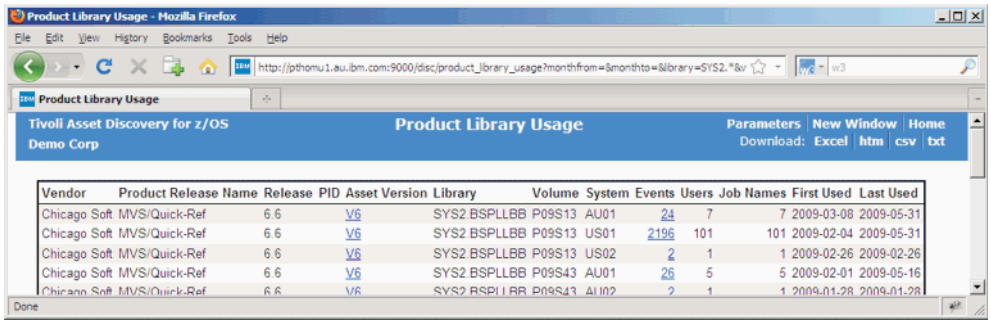


Table 51. Product Library Usage hyperlinks

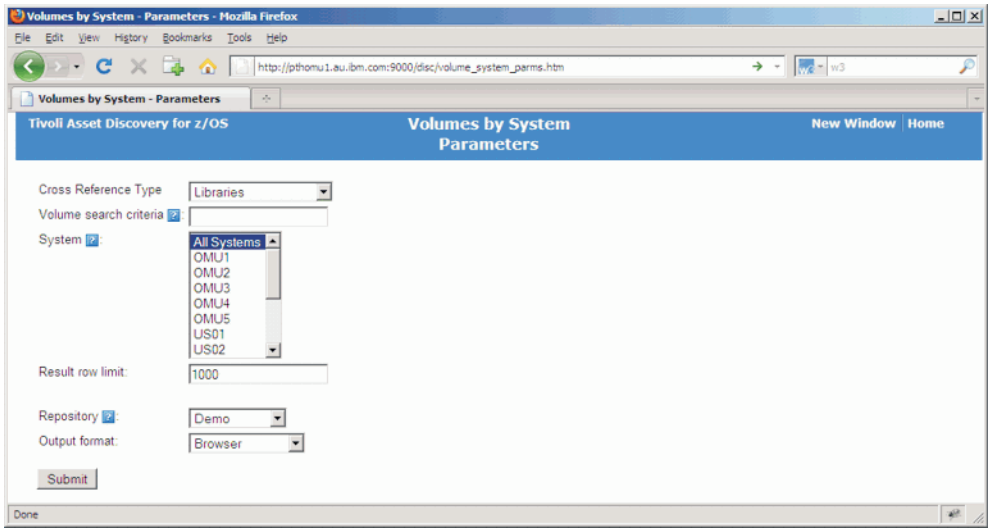
Column	Hyperlink
Asset Version	Drill down to Product Inventory
Events	Drill down to see module usage details

Volumes by System

Description: Summary of discovered library volumes by system

Batch name: /disc/volume_system

Sample parameters:



Sample output:

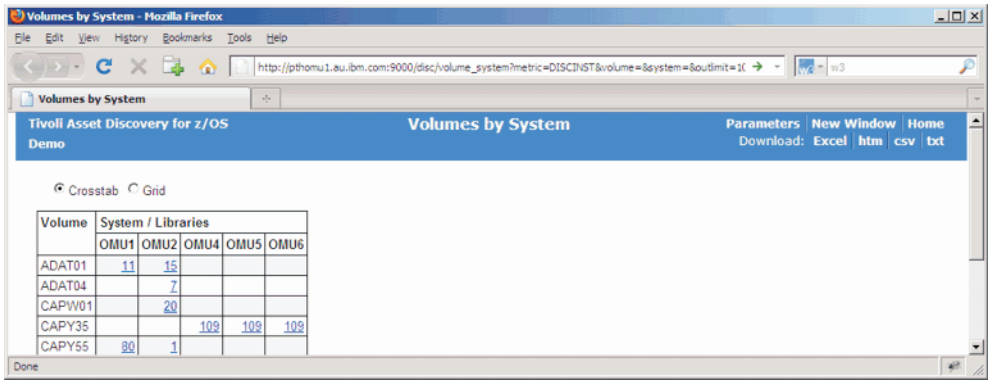


Table 52. Volumes by System hyperlinks

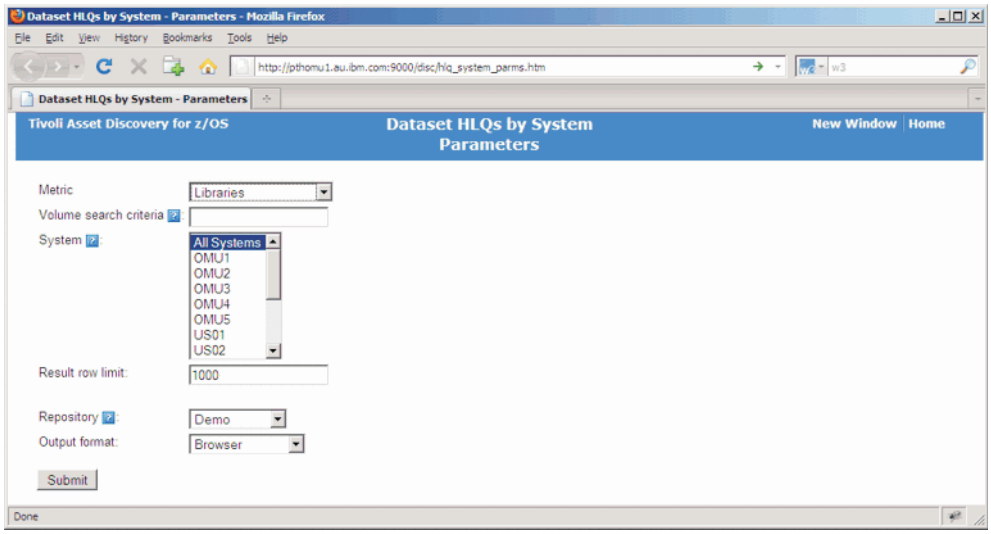
Column	Hyperlink
Libraries, Discovered Installed	Drill down to Search Libraries for volume and system

Data set HLQs by System

Description: Summary of discovered data set high-level qualifiers by system

Batch name: /disc/hlq_system

Sample parameters:



Sample output:

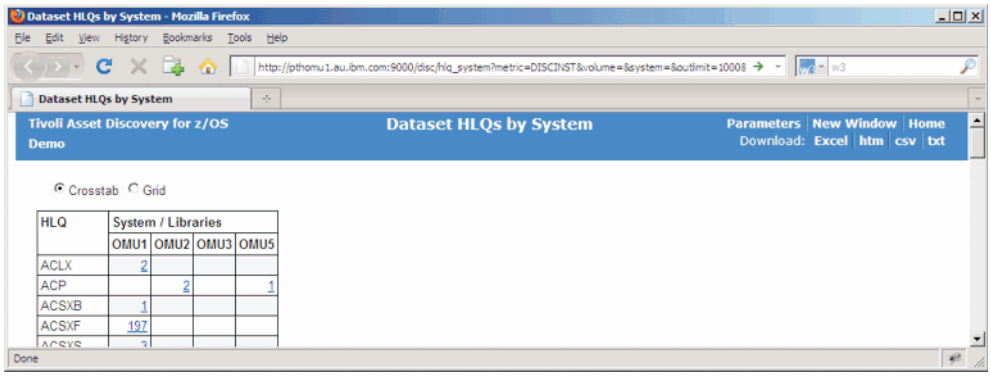


Table 53. Dataset HLQs by System hyperlinks

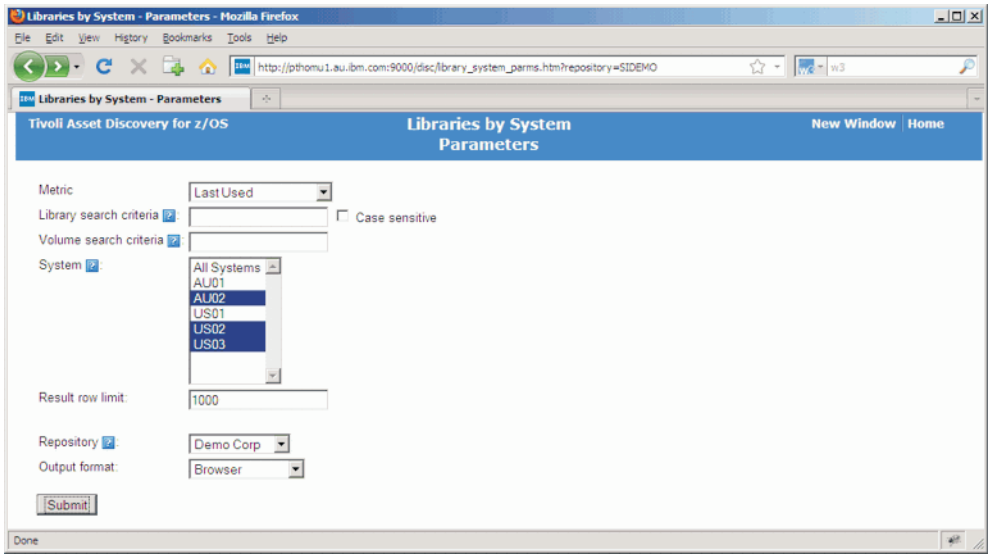
Column	Hyperlink
Libraries, Discovered Installed	Drill down to Search Libraries

Libraries by System

Description: Summary of discovered libraries by system

Batch name: /disc/library_system

Sample parameters:



Sample output:

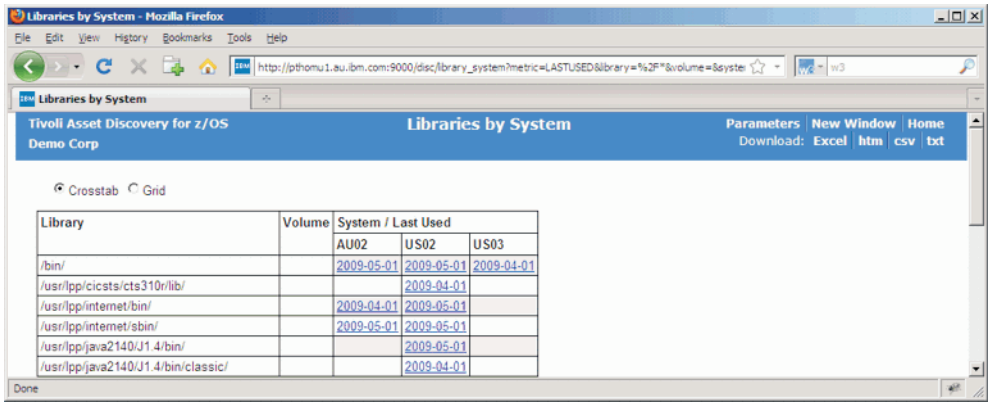


Table 54. Libraries by System hyperlinks

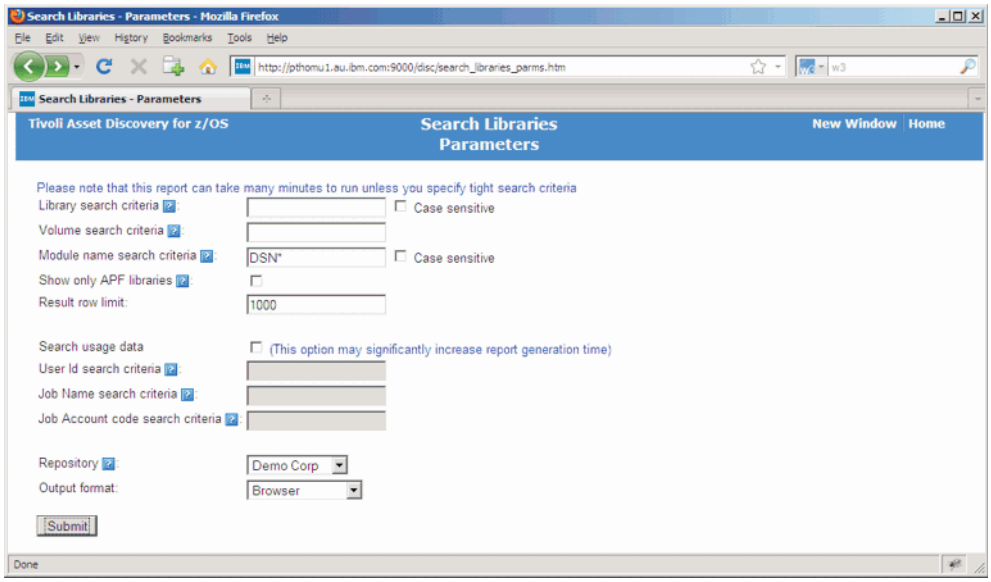
Column	Hyperlink
Libraries, Discovered Installed	Drill down to Search Libraries

Search Libraries

Description: Search Libraries, with optional filters for library name mask, and containing module name mask

Batch name: /disc/search_libraries

Sample parameters:



Sample output:

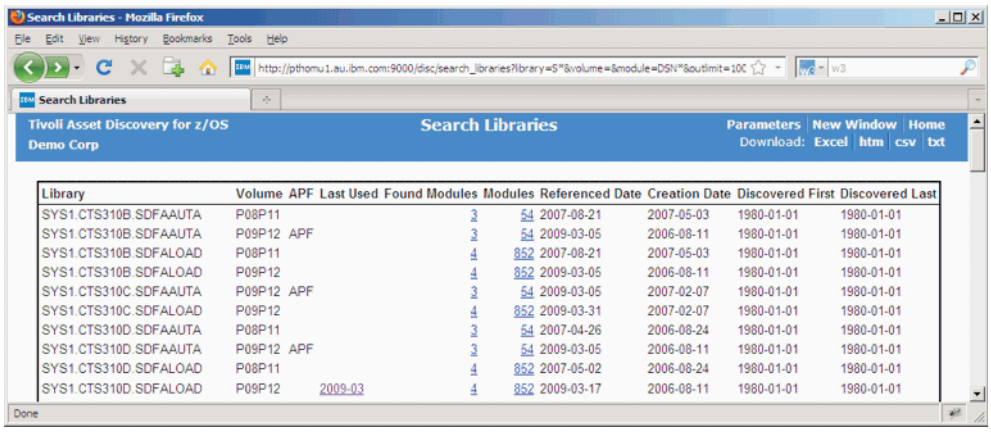


Table 55. Search Libraries hyperlinks

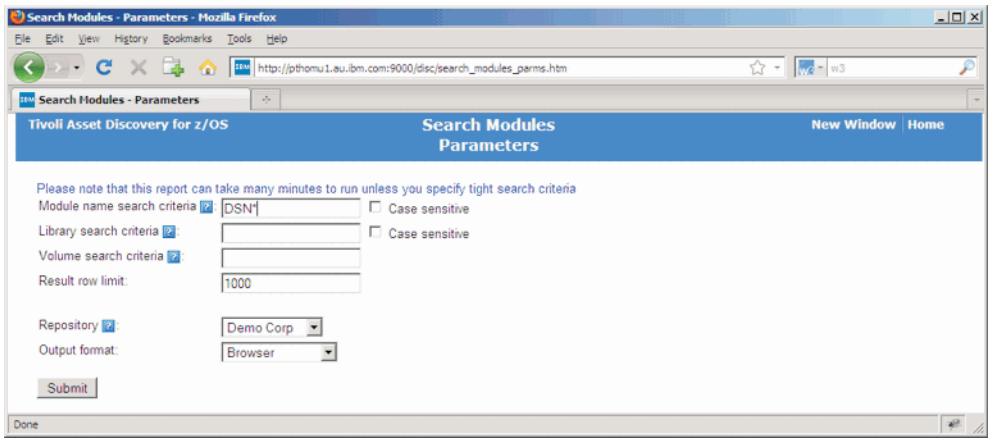
Column	Hyperlink
Last Used	Drill down to see module usage details
Found Modules	Drill down to Search Modules to see the found modules in the library
Modules	Drill down to Search Modules to see all modules in the library

Search Modules

Description: Search Modules with optional filters for module name mask and library name mask

Batch name: /disc/search_modules

Sample parameters:



Sample output:

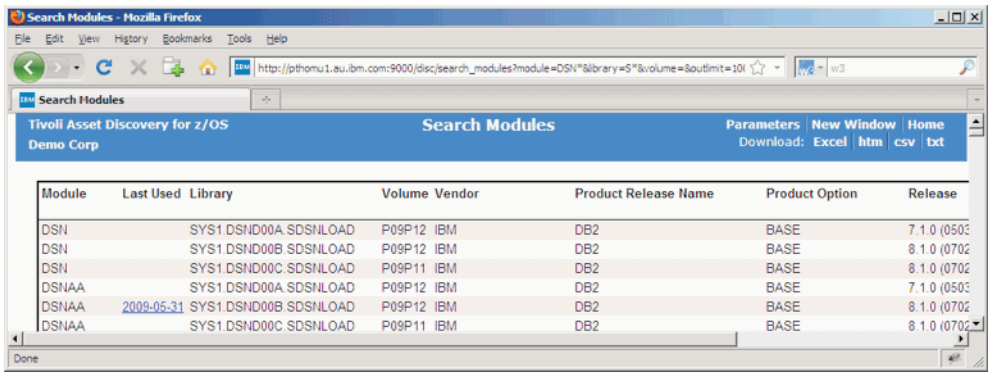


Table 56. Search Modules hyperlinks

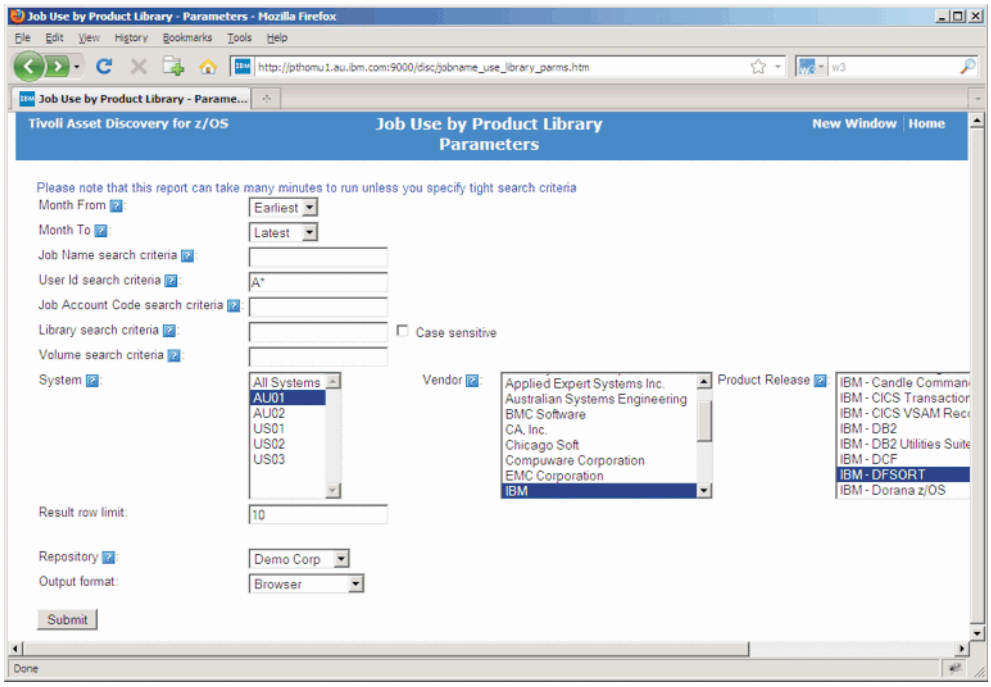
Column	Hyperlink
Last Used	Drill down to see module usage details

Job Use by Product Library

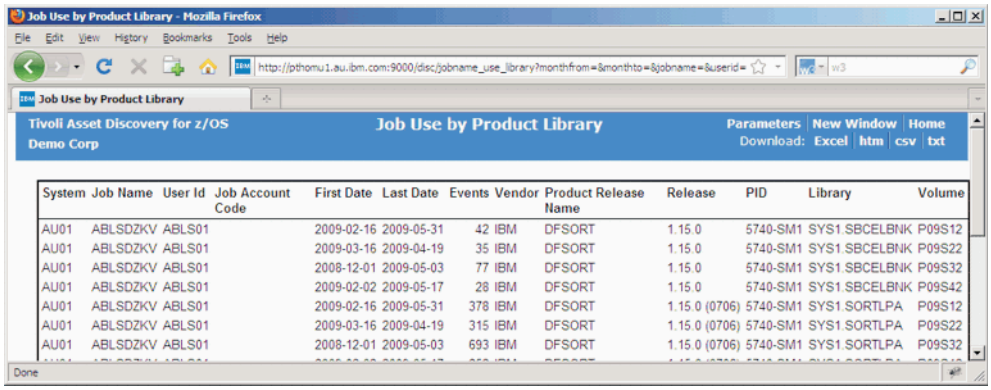
Description: Product release usage summary by Job name and Product Library

Batch name: /disc/jobname_use_library

Sample parameters:



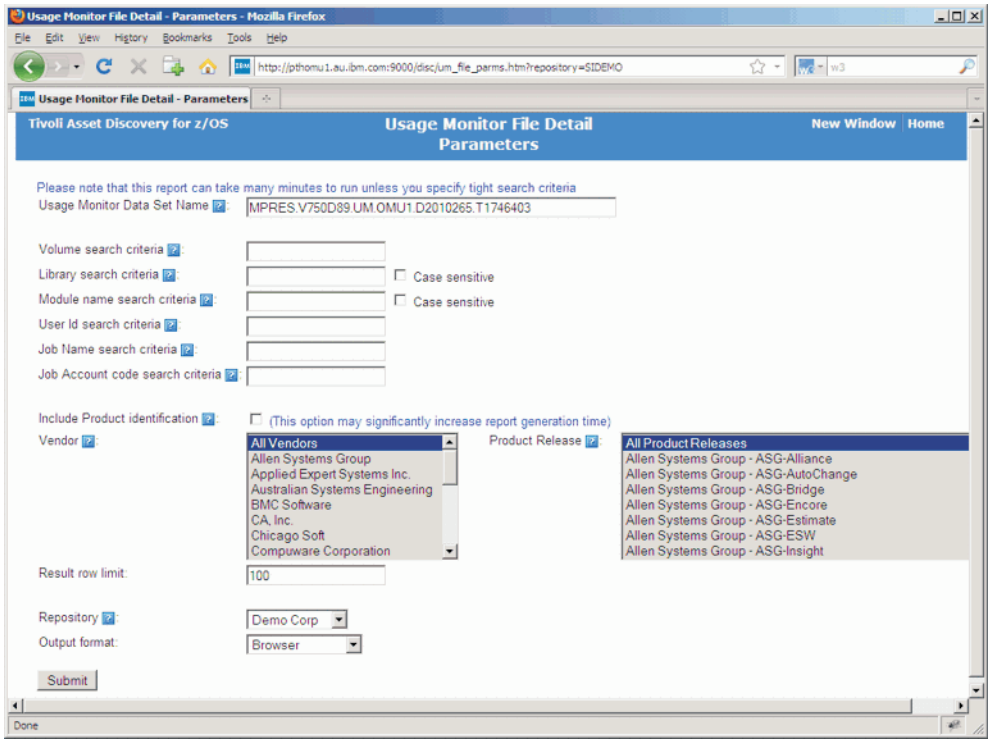
Sample output:

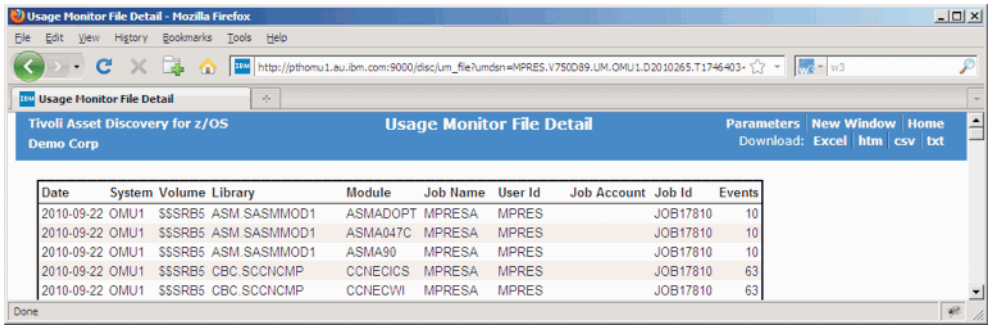


Usage Monitor File Detail

Description: Inspect usage detail in Usage Monitor raw data compressed files or ZCAT compressed files

Batch name: /disc/um_file





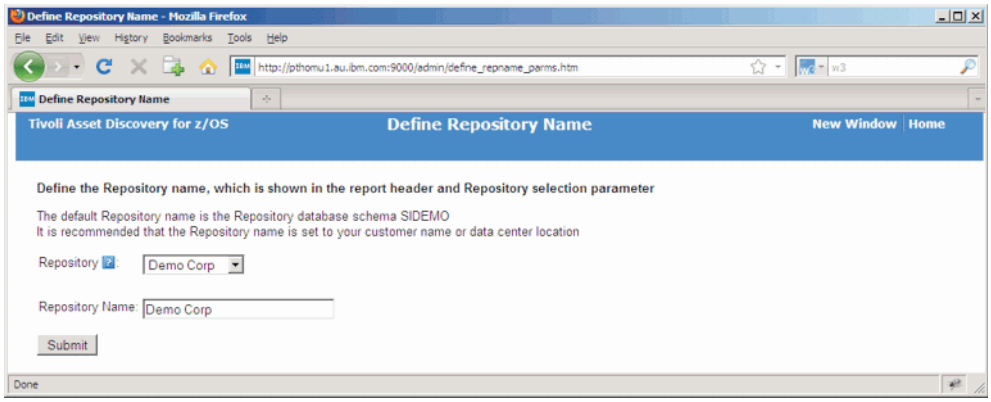
Analyzer administration

This section contains a list of the administration reports. For each report, two sample screens are provided. The first screen displays sample parameters, followed by sample output. Directly following these screens is a table which describes the function of each hyperlink in the output section of the report.

Define Repository Name

Description: Define the Repository name, which is shown in the report header and Repository selection parameter

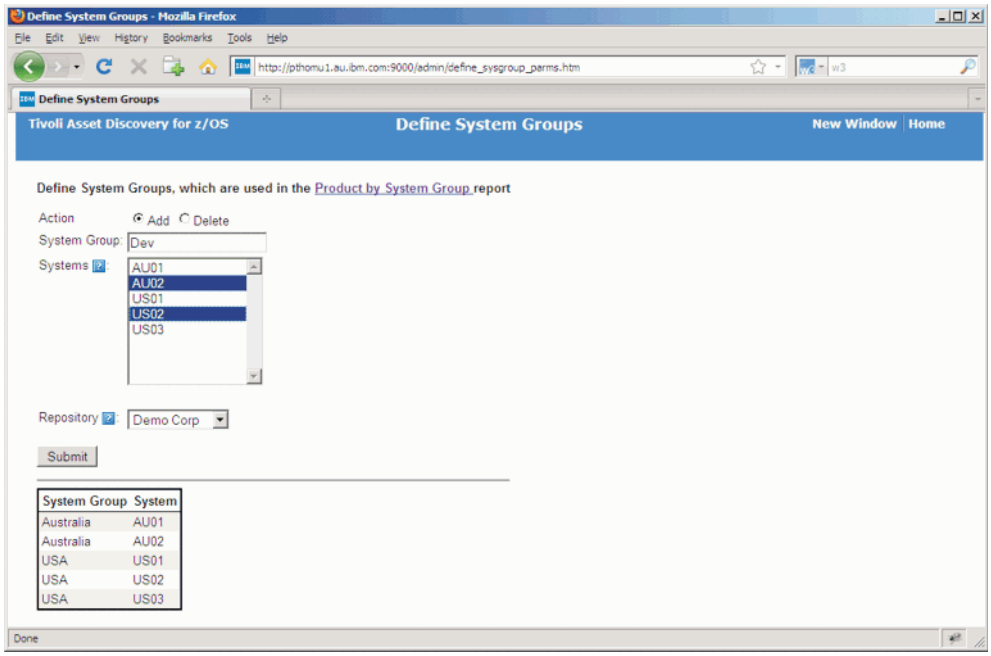
Sample parameters:



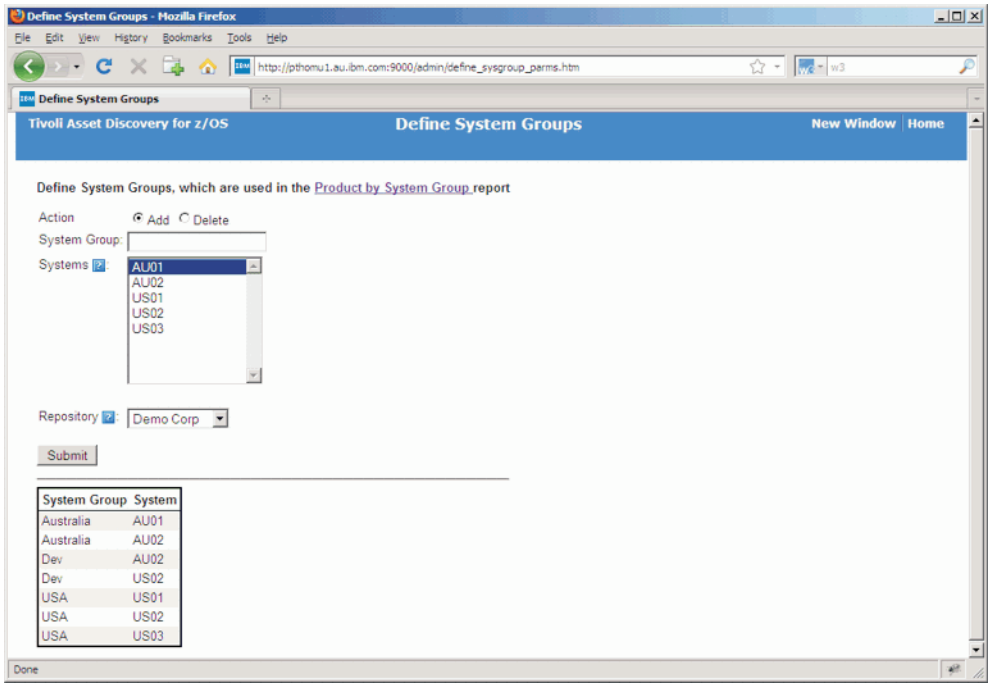
Define System Groups

Description: Define System Groups which are used in the Product by System Group report

Sample parameters:



Sample output:



Libraries with Unknown Modules

Description: Libraries with modules that have not been identified to a product release

Batch name: /disc/product_library_usage

This report shows cases where Tivoli Asset Discovery for z/OS cannot accurately identify a product release for a module. Common reasons why this occurs are:

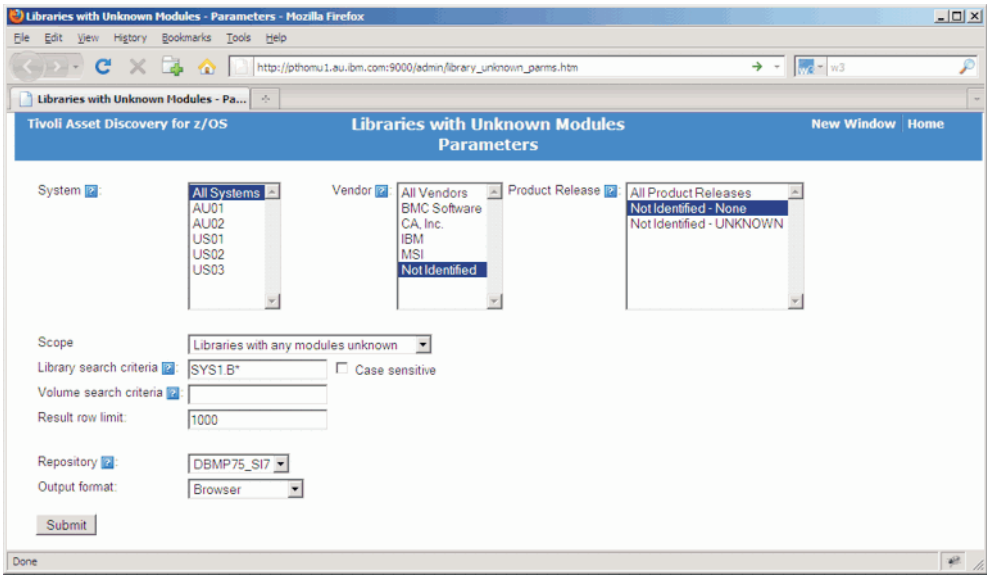
- The module is a customer developed application module, not a vendor product module.
- The library contains a small subset of product modules that have been copied from a product library.
- A new product release has been installed and it is not in the Global Knowledge Base.

In cases where the Match Engine cannot accurately identify a product release, it uses the following assignment logic:

1. IF at least 70% of the module names in the library match a known product, THEN
   - The Vendor, Product, and Option are set to the known product.
   - The Release is set to `UNKNOWN`.
2. ELSE the module name is in the Knowledge Base, THEN
   - The vendor is set to `Not identified`.
   - The Product, Option, and Release are set to `UNKNOWN`.
3. ELSE
   - The Vendor is set to `Not identified`.

- The Product, Option, and Release are set to None.

Sample parameters:



Sample output:

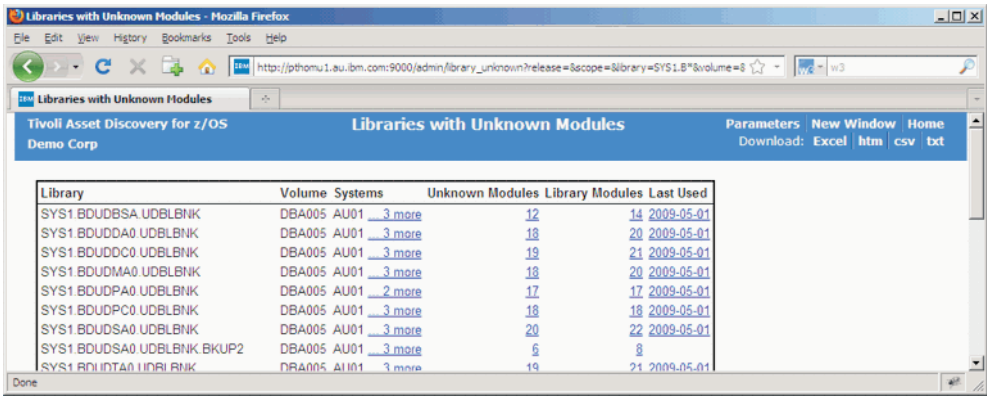


Table 57. Search Libraries hyperlinks

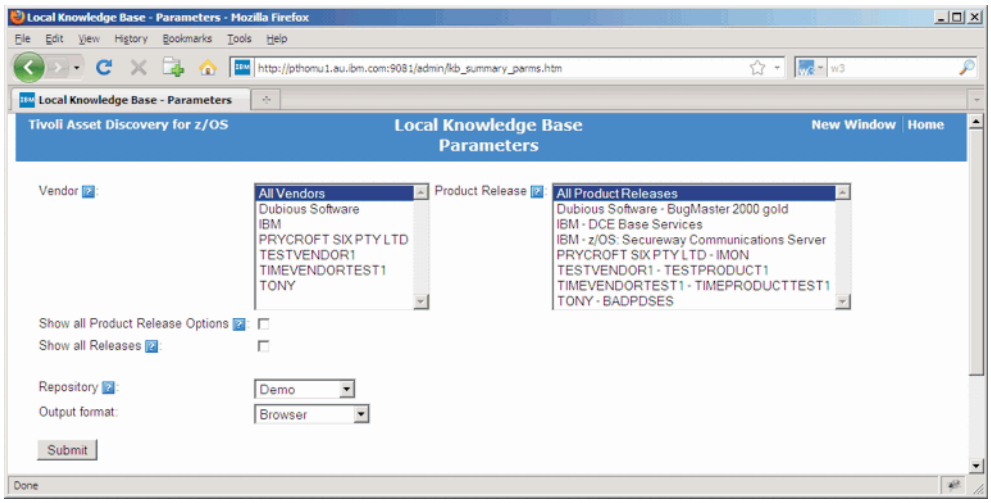
Column	Hyperlink
Systems	Drill down to Libraries with Unknown Modules showing all systems
Unknown Modules	Drill down to Search Modules to see all unknown modules in the library
Library Modules	Drill down to Search Modules to see all modules in the library
Last Used	Drill down to Product Library usage for unknown modules

LKB Summary

Description: Summary of products in the Local Knowledge Base catalog.

Batch name: /admin/lkb_summary

Sample parameters:



Sample output:

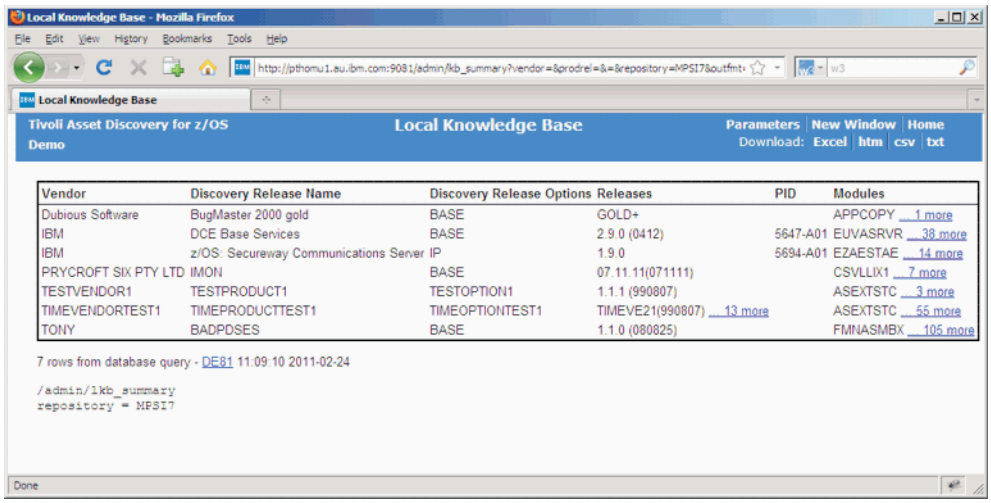


Table 58. Search Libraries hyperlinks

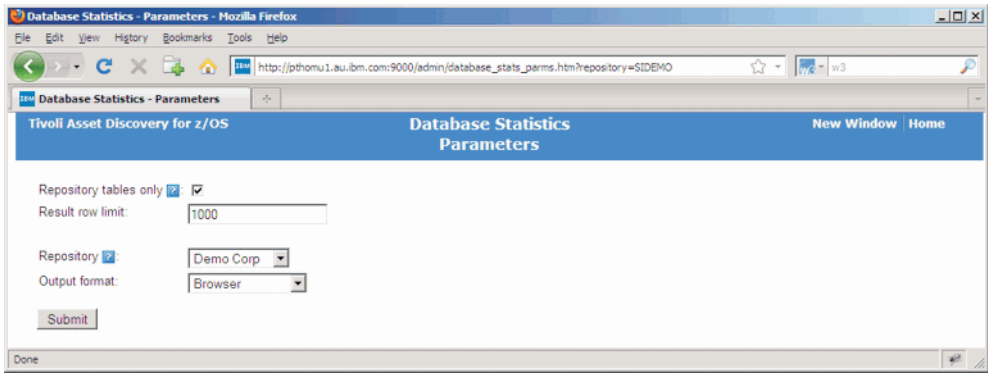
Column	Hyperlink
Releases	Drill down to LKB Summary to see all releases
Modules	Drill down to see all modules
Library Modules	Drill down to Search Modules to see all modules in the library
Last Used	Drill down to Product Library usage for unknown modules

Database Statistics

Description: Database statistics, including space used

Batch name: /admin/database_stats

Sample parameters:



Sample output:

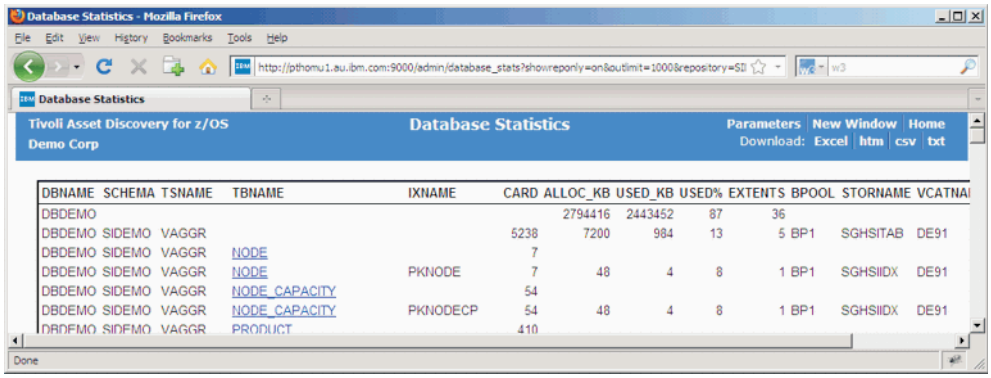


Table 59. Search Libraries hyperlinks

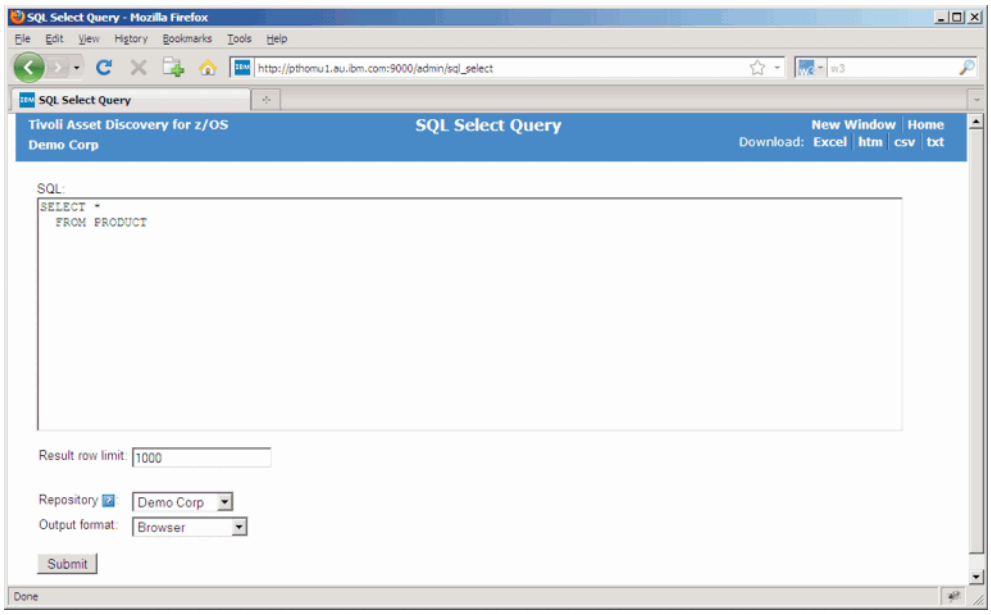
Column	Hyperlink
TBNAME	Drill down to browse table

SQL Select Query

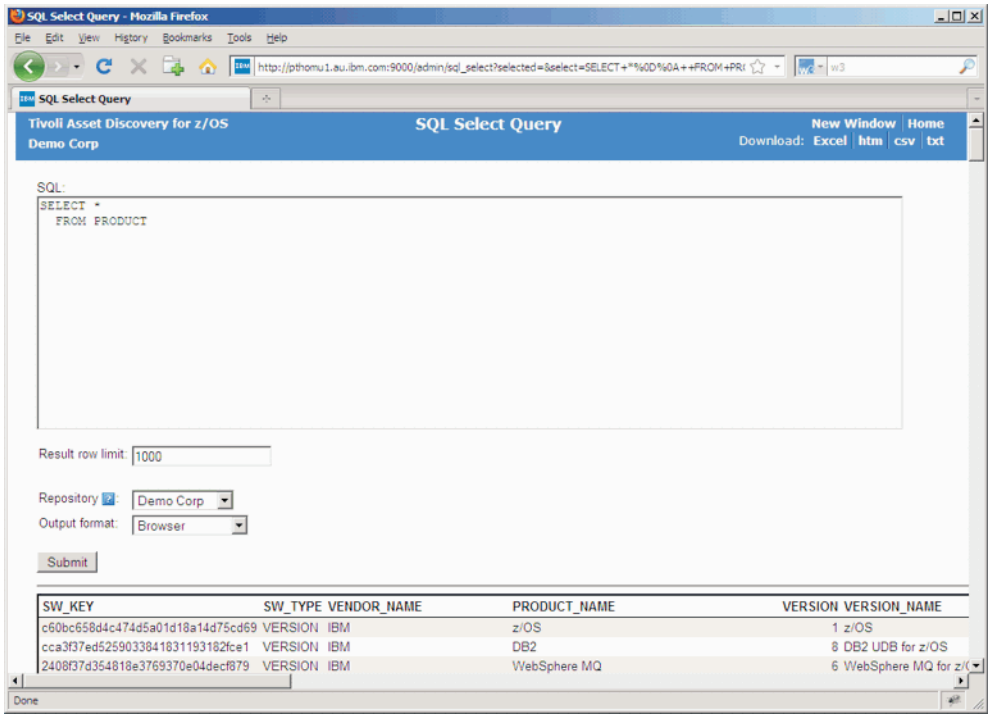
Description: Ad hoc SQL Select query.

Batch name: /admin/sql_select

Sample parameters:



Sample output:

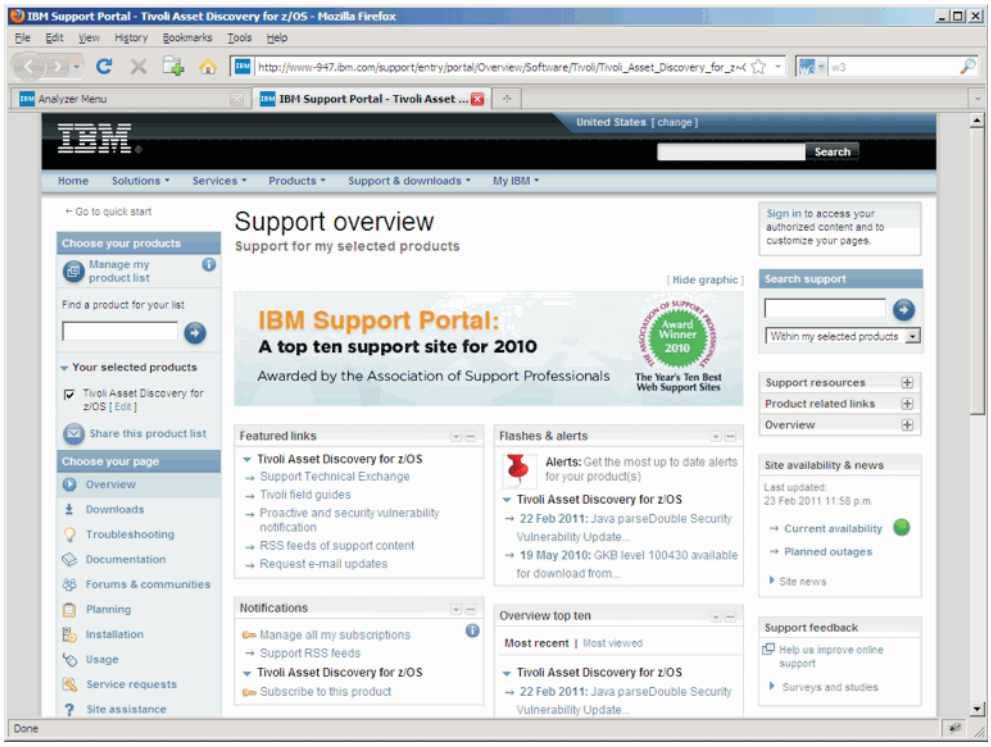


Support

Description: Tivoli Asset Discovery for z/OS Support Portal

URL: http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Asset_Discovery_for_z~OS

Sample output:

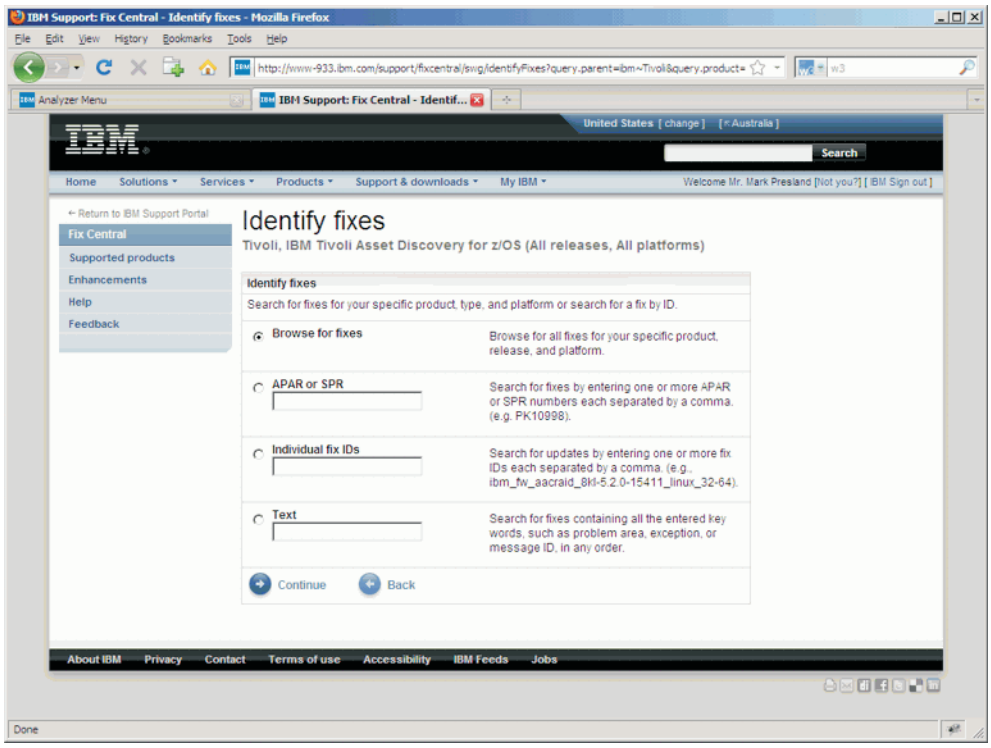


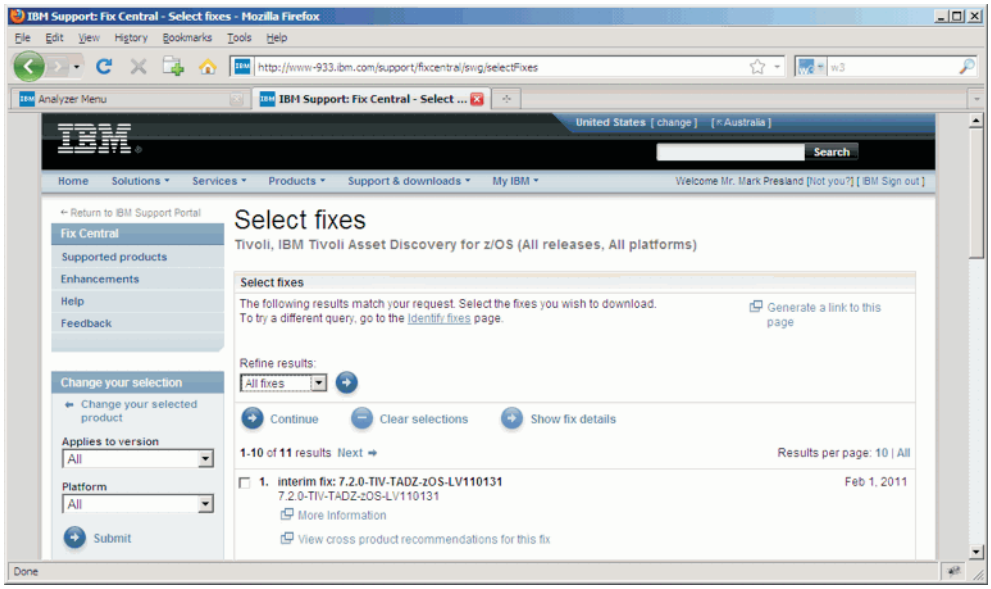
Download GKB

Description: Download the latest Global Knowledge Base from IBM Fix Central

URL: http://www.ibm.com/webapp/set2/common/login?rtn=fixcentral

Sample output:





Analyzer report parameters

This table describes all of the Analyzer Report parameters. See the Analyzer Reports section for which parameters are applicable to specific reports.

Table 60. Analyzer report parameters

On-line Mode Parameter	Batch Mode Parameter	Description
Exclude Unknown Releases	exclunknown=on	Exclude data that has not been identified to a product release. For example, customer applications
Include Product identification	inclident=on	Instead of only formatting the raw data, also query the database and show what product the module has been identified for
Job Account code search criteria	jobacc=...	Job Account Code search criteria
Job Name search criteria	jobname=...	Job Name search criteria
Library search criteria	library=...	Library search criteria
Machine	machine=...	Machine search criteria
Metric (Discovered Installed)	metric=DISCINST	Show Discovered Installed date
Metric (Module Events)	metric=EVENT	Show Module event count
Metric (Job Account Count)	metric=JAC	Show Job Account count
Metric (Job Name Count)	metric=JOBNAME	Show Job Name count
Metric (Last Used)	metric=LASTUSED	Show Last Used month
Metric (Libraries)	metric=LIBRARIES	Show library count
Metric (SCRT MSU)	metric=SCRT	Show SCRT MSU
Metric (User Id Count)	metric=USERID	Show User Id count
Module name search criteria	module=...	Module name search criteria
Month	month=...	Show data for specified month
Month From	monthfrom=...	Show data from specified month
Month To	monthto=...	Show data to specified month
MSU Type (Full Capacity MSU)	metric=MSU	Show Full Capacity MSU
MSU Type (Sub Capacity MSU)	metric=SUBCAPMSU	Show Sub-capacity MSU
Output format (Browser)	outfmt=	Output format is HTML with hyperlinks
Output format (Download CSV)	outfmt=csv	Output format is Comma Separated Value (CSV)
Output format (Download HTM)	outfmt=htm	Output format is HTML without hyperlinks
Output format (Download TXT)	outfmt=txt	Output format is Text line
Output format (Download XLS)	outfmt=xls	Output format is Excel 2002
Product	product=...	Asset Product search criteria
Product Release	prodrel=...	Discovery Product search criteria
Repository	repository=...	Repository
Repository Name	repname=...	Repository name
Repository tables only	showreponly=on	Show only tables for the Repository
Result row limit	outlimit=...	Limit the number of rows returned by the query
Scope (Libraries with any modules unknown)	scope=	Show libraries with any modules unknown
Scope (Libraries with all modules unknown)	scope=ALL	Show libraries with all modules unknown

Table 60. Analyzer report parameters (continued)

On-line Mode Parameter	Batch Mode Parameter	Description
Scope (Libraries with partial modules unknown)	scope=PARTIAL	Show libraries with partial modules unknown
Show all Asset Versions	showver=on	Show all Asset Versions
Show all Job Names	showalljobname=on	Show all Job Names
Show all Product Release Options	showoption=on	Show all Product Release Options
Show all Products	showallprod=on	Show all Products
Show all Releases	showrel=on	Show all Releases
Show all Systems	showallsys=on	Show all Systems
Show all User Ids	showalluserid=on	Show all User Ids
Show Asset Product Version Title	showvername=on	Show Asset Product Version Title
Show Features	showfeature=on	Show Features
Show Logical Partitions (LPARs)	showlpars=on	Show Logical Partitions (LPARs)
Show only APF libraries	apf=on	Show only APF libraries
Show only Vendors for discovered products	showdiscvendors=on	Show only Vendors for discovered products
Show Product Release Options	showoption=on	Show Product Release Options
Show product version title	showvername=on	Show product version title
SQL	select=...	SQL Select statement to issue
Status (Installed & Not used)	status=Installed & Not used	Audit report to include this status
Status (Installed & Used (history))	status=Installed & Used (history)	Audit report to include this status
Status (Installed & Used (last))	status=Installed & Used (last)	Audit report to include this status
Status (Uninstalled)	status=Un-installed	Audit report to include this status
Sysplex	sysplex=...	Sysplex search criteria
System	system=...	System search criteria
System Group	sysgroup=...	System Group search criteria
User Id search criteria	userid=...	User Id search criteria
Vendor	vendor=...	Vendor search criteria
Volume search criteria	volume=...	Volume search criteria

Analyzer report output columns

Table 61. Analyzer Report output columns

Column Name	Description
Asset Feature	Name of product version feature
Asset Product Name	Name of product version. This is usually the same as the official title. The name is normalized so that similar product versions are shown together when in a sorted list.
Asset Version	Version of product
Asset Version Title	Title of product version
APF	Indicates that the library was discovered as being defined to z/OS Authorized Program Facility

Table 61. Analyzer Report output columns (continued)

Column Name	Description
Creation Date	Create date of the library
Date	Date raw usage data was monitored
Discovered First	Date the resource was first discovered
Discovered Installed	Date the product was first discovered as being installed
Discovered Last	Date the resource was last discovered
Discovered Uninstalled	Date the product was discovered to be missing after previously been discovered as installed
Discovery Release Name	Name of discovered product release
Discovery Release Option	Name of discovered product option
Discovery Status	A value of “Found” means that the product has been discovered
End of Service	Date the vendor has said that the product release is no longer being supported
Events	The amount of times Usage Monitor has detected a module being loaded into storage for execution by an address space
EID	Feature Entitlement Identifier
Feature	Name of product version feature
First Date	First date in month that usage was detected
First Discovered	Date resource was first discovered
First Used	First date in month that usage was detected
Found Modules	Number of modules found for the search criteria
FMID	Product FMID
GKB Level	Global Knowledge Base level
HLQ	Data set High Level Qualifier
HW Model	Hardware Model
HW Name	Hardware Name
HW Plant	Hardware Plant
HW Serial	Hardware Serial number
HW Type	Hardware Type
Job Account	Job Account Code
Job Id	Last JES job id that usage was detected for the resource
Job Name	Job Name
Last Date	Last date in month that usage was detected
Last Job Id	Last JES job id that usage was detected for the resource
Last Used	Last date in month that usage was detected
Library	Library name. z/OS data set name or USS path name
Library Modules	Number of modules in the library
LPAR Name	Name of Logical Partition
LPAR Number	Number of Logical Partition
LPARs	Count of Logical Partitions

Table 61. Analyzer Report output columns (continued)

Column Name	Description
Machine	Machine
Max MSU	Maximum MSU capacity
Max SCRT-MSU	Maximum Subcapacity MSU
Module	Module name
Module Size	Module size
Modules	Count of modules
Month	Month of usage
MSU	Million of Service Units capacity
Observed Deleted	Date the resource was observed to be deleted
Observed First	Date the resource was first observed
Observed Last	Date the resource was last observed
Product	Name of Product
Product Modules	Count of product modules
Product Option	Product Release Option name
Product Release Name	Product Release name
Product Release Option	Product Release Option name
PID	Product Identifier
Referenced Date	Date library was last referenced
Release	Release
Repository	Repository name
S&S PID	Service Subscription Product Identifier
Status	Audit Trail status: • Installed and Not used • Installed and Used (history) • Installed and Used (last) • Uninstalled
Sysplex	Sysplex name
System	System name
System Group	System Group name
SCRT MSU	Sub-Capacity Reporting Tool MSU
Type	Node type: • • HW • • LPAR
Unknown Modules	Number of unknown modules
Usage Month	Month usage was detected
User Id	User id
User Name	User Name
Vendor	Vendor Name
Version	Product Version

Table 61. Analyzer Report output columns (continued)

Column Name	Description
Version Title	Product Version Title
Volume	Volume

Appendix C. Repository table layouts

Table 62. NODE

Column Name	Column Type	Column Length	Description
NODE_KEY	Char	32	Global Unique ID (GUID) for this entry
NODE_TYPE	Char	4	Entry Type: HW or LPAR
HW_TYPE	Char	4	System z Hardware Type, for example 2096
HW_MODEL	Char	3	System z Hardware Model, for example P03
HW_PLANT	Char	2	System z Hardware Plant, for example 02
HW_SERIAL	Char	12	System z Hardware Serial, for example. 000000013EED
HW_NAME	Char	10	Configured Hardware Name
HW_VENDOR	Char	10	System z Hardware Vendor, for example IBM
LPAR_NUMBER	Char	4	Logical Partition Number, for example 1
LPAR_NAME	Integer		Logical Partition Name, for example LPARSYS1
VMGUEST_NAME	Char	10	z/VM® Guest Name (if z/OS is running under z/VM)
HW_NODE_KEY	Char	32	NODE_KEY for related hardware parent
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated

Table 63. NODE_CAPACITY

Column Name	Column Type	Column Length	Description
NODE_KEY	Char	32	NODE GUID
PERIOD	Date		Month for this entry
START_TIME	Timestamp		First date that this entry is applicable for this Month
END_TIME	Timestamp		Last date that this entry is applicable for this Month
METRIC_TYPE	Char	10	Metric Type: MSU, SUBCAPMSTY
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated
QUANTITY	Integer		Metric Value

Table 64. PRODUCT

Column Name	Column Type	Column Length	Description
SW_KEY	Char	32	Global Unique ID (GUID) for this entry. For SW_TYPE=VERSION this will be the same value as VERSION_GUID For SW_TYPE=FEATURE this will be the same value as FEATURE_GUID
SW_TYPE	Char	8	Entry type - VERSION or FEATURE
VENDOR_NAME	Char	50	Vendor name

Table 64. PRODUCT (continued)

Column Name	Column Type	Column Length	Description
PRODUCT_NAME	Char	50	Product name, which is a normalized form of Version Name in order to group different versions of products under the same product name
VERSION	Integer		Version
VERSION_NAME	Char	50	Product Version Title
FEATURE_NAME	Char	50	Product Feature Title
PID	Char	16	Product Identifier
EID	Char	8	Entitlement Identifier for the Product Feature
SSPID	Char	8	Subscription & Support Product Identifier
SSEID	Char	8	Subscription & Support Entitlement Identifier for the Product Feature
PRICETYPE	Char	10	Price Type (not used in 7.2)
SUBCAPACITY	Char	20	IPLA Subcapacity type: Execution-based, Reference-based, z/OS-based, Not eligible, NULL
ICA	Char	1	Y or N: IBM Company Agreement license
IPLA	Char	1	Y or N: International Program License Agreement
VUE	Char	8	IPLA Value Unit Exhibit
VENDOR_GUID	Char	32	Globally Unique ID for VENDOR_NAME
PRODUCT_GUID	Char	32	Globally Unique ID for VENDOR_NAME + PRODUCT_NAME
VERSION_GUID	Char	32	Globally Unique ID for VENDOR_NAME + VERSION_NAME + VERSION
FEATURE_GUID	Char	32	Globally Unique ID for VENDOR_NAME + VERSION_NAME + VERSION + FEATURE_NAME
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated

Table 65. PRODUCT_INSTALL

Column Name	Column Type	Column Length	Description
SW_KEY	Char	32	Product GUID
SYSTEM_KEY	Char	32	System GUID
INSTALL_DATE	Date		Date the product was first observed to be installed on this System
UNINSTALL_DATE	Date		Date the product was first observed to be missing from this System
LAST_USED_DATE	Date		Date the product was last used on this System
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated

Table 66. PRODUCT_NODE_CAPACITY

Column Name	Column Type	Column Length	Description
SW_KEY	Char	32	Product GUID

Table 66. PRODUCT_NODE_CAPACITY (continued)

Column Name	Column Type	Column Length	Description
NODE_KEY	Date		Node GUID
PERIOD	Timestamp		Month for this entry
START_TIME	Timestamp		First date that this entry is applicable for this Month
END_TIME	Timestamp		Last date that this entry is applicable for this Month
METRIC_TYPE	Integer		Metric Type: INSTALLED, JOBNAMES, MODULES, USERS, SUBCAPMSU
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated
QUANTITY	Float		Metric Value

Table 67. PRODUCT_USE

Column Name	Column Type	Column Length	Description
PERIOD	Date		Month for this entry
SYSTEM_KEY	Char	32	System GUID
SW_KEY	Char	32	Product GUID
FLPARID	Integer		TLPAR.FLPARID for convenient linking with PRODUCT_USE_DETAIL
HW_NODE_KEY	Char	32	NODE GUID for Hardware NODE that this System was last running on in this month
USER_CNT	Integer		MAX distinct Userid count
JOBNAME_CNT	Integer		MAX distinct Job Name count
ACCOUNT_CNT	Integer		MAX distinct Account Code count
SCRT_MSU			Sub-capacity Reporting Tool MSU (millions of service units per hour)
EVENT_CNT	Double		SUM of Module usage
START_DATE	Date		Date within this Period that usage was for first recorded
END_DATE	Date		Date within this Period that usage was for last recorded
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated

Table 68. PRODUCT_USE_DETAIL

Column Name	Column Type	Column Length	Description
PERIOD	Date		Month for this entry
FLPARID	Integer		System TLPAR.FLPARID for convenient linking with PRODUCT_USE
VERSION_GUID	Char	32	Product Version GUID
FEATURE_GUID	Char	32	Product Feature GUID
USERNAME	Char	8	User ID
JOBNAME	Char	8	Job Name
ACCOUNTCODE	Char	20	First 20 chars of the Job Account Code
EVENT_CNT	Double		SUM of Module usage

Table 68. PRODUCT_USE_DETAIL (continued)

Column Name	Column Type	Column Length	Description
START_DATE	Date		Date within this Period that usage was for first recorded
END_DATE	Date		Date within this Period that usage was for last recorded

Table 69. SYSTEM

Column Name	Column Type	Column Length	Description
SYSTEM_KEY	Char	32	Global Unique ID (GUID) for this entry
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated
SID	Char	4	Product system ID. By default this is the SMFID. In cases where the same SMFID is used on different systems, the SID must be defined to a unique value for the customer enterprise in the Usage Monitor
SMFID	Char	4	z/OS SMF ID
SYSPLEX	Char	8	z/OS Sysplex name

Table 70. SYSTEM_NODE

Column Name	Column Type	Column Length	Description
SYSTEM_KEY	Char	32	System GUID
NODE_KEY	Char	32	Node GUID
PERIOD	Date		Month this entry is for
START_TIME	Timestamp		Time it was first observed that this system is using this Node in this month period
END_TIME	Timestamp		Time it was last observed that this system is using this Node in this month period
LAST_UPDATE_TIME	Timestamp		Time stamp entry was last updated

Table 71. TACCOUNT

Column Name	Column Type	Column Length	Description
FACCOUNTID	Integer		Account ID
FACCOUNTCODE	Char	20	Job Account Code, truncated to 20 characters

Table 72. TINVCTL

Column Name	Column Type	Column Length	Description
FINVID	Integer		Inventory ID
FINVNAME	Char	24	Inventory ID name
FVERSIONGKBID	Char	15	Version of the Global KB used for matching
FIQDATE	Timestamp		Current® date that the Inquisitor was run on the mainframe
FIQDATEFIRST	Timestamp		Date the Inquisitor was first run on the mainframe
FMIGDATEFIRST	Timestamp		Initial migrate date

Table 72. TINVCTL (continued)

FMIGDATE	Timestamp		Date Inquisitor data was loaded to Repository
FSYSCPUID	Char	12	CPU Serial number
FSYSPLEXID	Char	8	Sysplex name
FSYSFMID	Char	8	System FMID
FSMFID	Char	20	SMF ID
FPREVINVID	Integer		Predecessor Inventory ID
FIQSYSPLEXUSE	Smallint		Was the SYSPLEX option turned on?
FVENDORCNT	Integer		Number of Vendors in Inventory
FPRODUCTCNT	Integer		Total Product count in Inventory ID
FLIBCNT	Integer		Total number of libraries in the inventory
FINVCONT	Char	1	Continuous Inventory
FINVTYPE	Smallint		Type of Inventory
FIQNAME	Varchar	254	Inquisitor Filename and path for DB2 schema name

Table 73. TINVREG

Column Name	Column Type	Column Length	Description
FREGIONID	Integer		Parent Region ID
FINVID	Integer		Inventory ID

Table 74. TIQHISTORY

Column Name	Column Type	Column Length	Description
FHISTORYID	Integer		Unique ID
FINVID	Integer		Inventory ID
FIQDATE	Timestamp		Inquisitor date
FMIGTYPE	Smallint		Migrate type
FIPADDR	Varchar	254	List of IP Addresses
FSUBNETMASK	Varchar	254	List of Subnet Masks
FMACADDR	Varchar	254	List of MAC Addresses
FOSNAME	Varchar	254	Operating System Name
FOSTYPE	Varchar	254	OS type e.g. "WINNT"
FOSVERSION	Varchar	254	Operating System Version
FOSBUILDNUMBER	Varchar	254	Build Number of OS
FOSCSDVERSION	Varchar	254	Latest Service Pack Installed
FOSSERNUM	Varchar	254	OS product serial identification number.
FOSINSTDATE	Timestamp		Date of OS installation. Null if unknown.
FOSLASTBOOTDATE	Timestamp		Date of OS last boot. Null if unknown.
FOSLASTBOOTDATE	Timestamp		Date of OS last boot. Null if unknown.
FTOTALMEMORY	Integer		Total Physical Memory in MB
FVERSIONGKB	Char	15	The Version the of the GKB the IQ was matched with.

Table 75. TJOBDATA

Column Name	Column Type	Column Length	Description
FJOBNAME	Char	8	Job Name
FJOBID	Integer		Job ID
FJOBTYPE	Char	6	Job Type

Table 76. TLIBRARY

Column Name	Column Type	Column Length	Description
FLIBID	Integer		Library ID
FLIBNAME	Char	128	Library name
FINVID	Integer		Inventory ID
FCREATIONDATE	Timestamp		Library creation date on Mainframe
FLIBDEVNUM	Char	4	DASD device number
FREFERENCEDATE	Timestamp		Date library last referenced
FLIBVOLSER	Char	8	Volser library resides on
FTRACKSALLOC	Char	10	Number of allocated tracks
FTRACKSUSED	Char	10	Number of used tracks
FORIGIN	Char	1	Blank - PDS, E - PDSE, V - VTOC
FCATALOG	Char	1	S - SMS managed, C - Cataloged, U uncataloged W - cataloged on wrong volume
FLINKLIST	Char	1	Is this a link listed library?
FLINKPACK	Char	1	Is this library in the Linkpack
FAPFAUTH	Char	1	Is this library APF authorized
FLASTUSAGE	Date		1st month of the most recent usage applied to any module in this library
FUSEFLAG	Smallint		Flag for library usage
FMODCNT	Integer		Number of modules in library
FOBSERVEFIRST	Timestamp		Date and time that library was first observed
FOBSERVELAST	Timestamp		Date and time that library was last observed
FOBSERVEDELETED	Timestamp		Date and time that library was deleted from Inquisitor data.
FCHECKSUM	Char	40	A checksum of module names and sizes in a given library used to determine whether a library has changed.
FSTORAGEGROUP	Char	8	The storage group the library belongs to

Table 77. TLIBSYS

Column Name	**Column Type**	**Column Length**	**Description**
FLIBID	Integer		Library ID
FLPARID	Integer		LPAR ID
FOBSERVEFIRST	Timestamp		Date and time that library was first observed
FOBSERVELAST	Timestamp		Date and time that library was last observed
FOBSERVEDELETED	Timestamp		Date and time that library was deleted from Inquisitor data

Table 78. TLPAR

Column Name	Column Type	Column Length	Description
FLPARID	Integer		LPAR ID
FLPARNAME	Char	20	Name of the LPAR
FUSEFLAG	Smallint		Indicates if usage has been attributed to this LPAR
FEDITFLAG	Smallint		Has this LPAR record been updated manually
FMANF	Char	10	Machine manufacturer
FMACHINE	Char	12	CPU Model
FSERIALNO	Char	12	CPU Serial number
FSYSPLEXID	Char	8	Sysplex name if in a Sysplex
FMIPS	Integer		Number of MIPS for LPAR

Table 79. TMODULE

Column Name	Column Type	Column Length	Description
FMODID	Integer		Module ID
FMODNAME	Char	40	Module name
FLIBID	Integer		Library ID
FPOVLIBID	Integer		Product Library ID
FMODFLAG	Smallint		Module indication flag as to which product version it belongs to and whether it has been superseded.
FFMID	Char	12	FMID
FMODSIZE	Char	8	Module size
FUSEFLAG	Smallint		Flag for module usage
FMODTYPE	Smallint		Type of module
FOBSERVEFIRST	Timestamp		Date and time that module was first observed
FOBSERVELAST	Timestamp		Date and time that module was last observed
FOBSERVEDELETED	Timestamp		Date and time that module was deleted from Inquisitor data

Table 80. TPARAM

Column Name	Column Type	Column Length	Description
FKEY	Char	64	Parameter Key
FVALUE	Char	254	Parameter Value

Table 81. TPERIODS

Column Name	Column Type	Column Length	Description
FPERIOD	Date		Calendar month for usage
FINVID	Integer		Inventory ID
FSUMMARISED	Smallint		Summary status

Table 82. TPOVINV

Column Name	Column Type	Column Length	Description

Table 82. TPOVINV (continued)

FPOVINVID	Integer		Unique ID
FPOVID	Integer		Product ID
FINVID	Integer		Inventory ID
FPOVGID	Integer		Global Knowledge Base Version ID
FOBSERVEFIRST	Timestamp		First time observation was made
FOBSERVELAST	Timestamp		Last time observation was made
FOBSERVEDELETED	Timestamp		First time observation was not found in this library
FPRODUCTID	Integer		Product ID
FVENDORID	Integer		Vendor ID
FPRODINVID	Integer		Product and Inventory ID
FPATCHLIST	Varchar	254	List of current patches applied to z/OS UNIX product.

Table 83. TPOVLIB

Column Name	Column Type	Column Length	Description
FPOVLIBID	Integer		Unique ID
FPOVINVID	Integer		Product inventory ID
FLIBID	Integer		Library ID
FPOVLIBPID	Integer		Previous Product Version ID
FMATCHCODE	Char	3	Matching code
FMATCHID	Integer		Link to Inquisitor Decision table
FPRODUCTPCT	Integer		Product percentage used during match
FVERSIONPCT	Integer		Version percentage used during match
FOBSERVEFIRST	Timestamp		First time observation was made
FOBSERVELAST	Timestamp		Last time observation was made
FOBSERVEDELETED	Timestamp		First time observation was not found in this library
FMODCNT	Integer		Number of load modules in library for product

Table 84. TPRODUCT

Column Name	Column Type	Column Length	Description
FPRODUCTID	Integer		Product ID
FPRODUCTNAME	Char	50	Product Name (could be alias name)
FGLOBALNAME	Char	50	Product Name (always Global Name if Alias is used)
FOPTIONNAME	Char	30	Option Name
FVENDORID	Integer		Vendor ID
FPRODSTATUS	Smallint		Billable Status
FCATEGORY	Char	30	Product Category
FDESCRIPTION	Varchar	254	Product Description

Table 85. TREGCLASS

Column Name	Column Type	Column Length	Description
FREGCLASSID	Smallint		Region Class ID
FCLASSNAME	Char	24	Region classification title
FICONID	Smallint		Icon Lookup

Table 86. TREGION

Column Name	Column Type	Column Length	Description
FREGIONID	Integer		Region ID
FREGIONNAME	Char	64	Region Name
FPARENTID	Integer		Owning Region ID
FREGCLASSID	Smallint		Region category
FSEQ	Smallint		Sequence number for regions

Table 87. TREGLEAF

Column Name	Column Type	Column Length	Description
FPARENTID	Integer		Parent Region ID
FREGIONID	Integer		Region ID

Table 88. TUIMPORTCTRL

Column Name	Column Type	Column Length	Description
FINVID1	Integer		Primary Inventory ID
FINVID2	Integer		Second ID
FINVID3	Integer		Third ID
FINVID4	Integer		Fourth ID
FINVID5	Integer		Fifth ID
FINVID6	Integer		Sixth ID
FINVID7	Integer		Seventh ID
FINVID8	Integer		Eighth ID
FMODVPOV	Char	1	If non 0 allows relaxed VPOV assignment
FLPARNAME	Char	20	LPAR name

Table 89. TUSELIB

Column Name	Column Type	Column Length	Description
FUSELIBID	Integer		Unique ID
FLPARID	Integer		LPAR ID
FLIBID	Integer		Library ID

Table 90. TUSEMTD

Column Name	Column Type	Column Length	Description
FMTDID	Integer		Unique ID
FLPARID	Integer		LPAR ID
FMODID	Integer		Module ID

Table 90. TUSEMTD (continued)

FJOBID	Integer		Job ID
FUSERID	Integer		User ID
FPOVLIBID	Integer		Product Library ID
FEVENTCNT	Float		Total calls to module for this month
FPERIOD	Date		Calendar month that usage occurred
FFIRSTDATE	Date		First day of usage in the month
FLASTDATE	Date		Last day of usage in the month
FPROVIDER	Char	4	Provider Service
FPOVINVID	Integer		Unique ID
FPRODINVID	Integer		Product and Inventory ID
FACCOUNTID	Integer		Account ID
FJESID	Char	8	Last JES job ID updated for the month

Table 91. TUSEPO

Column Name	Column Type	Column Length	Description
FUSEPOVINVID	Integer		Unique ID
FLARPID	Integer		LPAR ID
FPRODINVID	Integer		Product & inventory ID
FJOBCNT	Integer		Number of distinct Jobs for a product
FUSERCNT	Integer		Number of distinct Users for a product
FEVENTCNT1	Float		Sum of calls to this product current month
FEVENTCNT3	Float		Sum of calls to this product previous 3 month
FEVENTCNT6	Float		Sum of calls to this product previous 4-6 month
FEVENTCNT9	Float		Sum of calls to this product previous 7-9 month
FEVENTCNT12	Float		Sum of calls to this product previous 10-12 month
FPERIOD	Date		Calendar month in which usage occurred
FFIRSTUSED	Date		The earliest usage date in month
FLASTUSED	Date		The most recent usage date in month

Table 92. TUSEPOV

Column Name	Column Type	Column Length	Description
FUSEPOVINVID	Integer		Unique ID
FLARPID	Integer		LPAR ID
FPOVINVID	Integer		POVINV ID
FJOBCNT	Integer		Number of distinct Jobs for a product
FUSERCNT	Integer		Number of distinct Users for a product
FEVENTCNT1	Float		Sum of calls to this product current month
FEVENTCNT3	Float		Sum of calls to this product previous 3 month

Table 92. TUSEPOV (continued)

FEVENTCNT6	Float		Sum of calls to this product previous 4-6 month
FEVENTCNT9	Float		Sum of calls to this product previous 7-9 month
FEVENTCNT12	Float		Sum of calls to this product previous 10-12 month
FPERIOD	Date		Calendar month in which usage occurred
FFIRSTUSED	Date		The earliest usage date in month
FLASTUSED	Date		The most recent usage date in month
FPRODINVID	Integer		Product Inventory ID
FSEQNO	Smallint		Internal use only
FACCCNT	Integer		Number of distinct account codes

Table 93. TUSEPOVLIB

Column Name	Column Type	Column Length	Description
FUSEPOVLIBID	Integer		Unique ID
FLARPID	Integer		LPAR ID
FPOVLIBID	Integer		POVLIB ID
FJOBCNT	Integer		Number of distinct Jobs for a product
FUSERCNT	Integer		Number of distinct Users for a product
FEVENTCNT1	Float		Sum of calls to this product current month
FEVENTCNT3	Float		Sum of calls to this product previous 3 month
FEVENTCNT6	Float		Sum of calls to this product previous 4-6 month
FEVENTCNT9	Float		Sum of calls to this product previous 7-9 month
FEVENTCNT12	Float		Sum of calls to this product previous 10-12 month
FPERIOD	Date		Calendar month in which usage occurred
FFIRSTUSED	Date		The earliest usage date in month
FLASTUSED	Date		The most recent usage date in month
FSEQNO	Smallint		Internal use only
FACCCNT	Integer		Number of distinct account codes

Table 94. TUSEPRS

Column Name	**Column Type**	**Column Length**	**Description**
FUSEPRSID	Integer		Unique ID for TUSEPRS table
FREGVEND	Char	16	Product Registration Vendor name
FREGPROD	Char	16	Product Registration Product name
FREGFEAT	Char	16	Product Registration Feature name
FREGVRN	Char	6	Product Registration Version
FREGPID	Char	8	Product Registration Product identifier
FREGFLAGS	Char	8	Product Registration flags

Table 94. TUSEPRS (continued)

Column Name	Column Type	Column Length	Description
FLPARID	Integer		LPAR ID
FPERIOD	Date		Calendar month when usage occurred
FFIRSTDATE	Date		The earliest usage date in month
FLASTDATE	Date		The most recent usage date in month

Table 95. TUSERDATA

Column Name	Column Type	Column Length	Description
FUSERID	Integer		User ID
FUSERNAME	Char	10	User Name
FORGNAME	Char	8	Owning Organization
FREALNAME	Char	20	Real person's name

Table 96. TVENDOR

Column Name	Column Type	Column Length	Description
FVENDORID	Integer		Vendor ID
FVENDORNAME	Char	50	Vendor Name (could be alias name)
FGLOBALNAME	Char	30	Vendor Name (always Global Name if Alias is used)
FVENDORGUID	Char	32	Vendor globally unique ID

Table 97. TVERSION

Column Name	Column Type	Column Length	Description
FPOVID	Integer		Version ID
FVERSIONNAME	Char	44	Version name
FPPNUMNAME	Char	16	PPNUM
FPRODUCTID	Integer		Product Option ID
FMINUSAGE	Float		Minimum usage threshold
FVERSIONGUID	Char	32	Product version globally unique identifier. PRODUCT.SW_KEY for SW_TYPE = 'VERSION'
FFEATUREGUID	Char	32	Product feature globally unique identifier. PRODUCT.SW_KEY for SW_TYPE = 'FEATURE'

Appendix D. Performance and tuning

Initial space allocation

This section is useful for the database administrator who must determine space requirements for Tivoli Asset Discovery for z/OS. Listed in the table Table 98 are guidelines for the initial spaces allocation based on the number of LPARs. The value for the SIZE parameter is specified in HSISCUST.

Table 98. Initial space allocation for the product

SIZE=	Initial space allocation	Number of LPARs
1	2200 cylinders	1-10
2	5500 cylinders	11-20
3	17500 cylinders	>20

Table 99. Initial space allocations for the 4 largest tables

SIZE=	WMZIQTS (Inquisitor modules)	VMODULE (Repository modules)	VUSEMTD (Repository usage records)	VPRODDET (Repository Product Use Detail Records)
1	504,000 KB for 1,000,000 modules	120,960 KB for 1,000,000 modules	72,000 KB for 1,000,000 records	13,680 KB for 100,000 records
2	1,260,000 KB for 2,5000,000 modules	302,400 KB for 2,500,000 modules	432,000 KB for 6,000,000 records	68,400 KB for 500,000 records
3	3,528,000 KB for 7.000,000 modules	846,720 KB for 7,000,000 modules	1,800,000 KB for 25,000,000 records	684,000 KB for 5,000,000 records

Note: For some sites, table space VUSEMTD can be large. For performance and space management requirements, you should consider defining the table space as a partitioned table space.

Choosing a DB2 subsystem for this product

The DB2 resources required for this product do not need to be defined in a production DB2 subsystem in order to minimize competition for mainframe resources in the DB2 production environments. To avoid competing for mainframe resources, run the jobs for the Inquisitor Import and Usage Import during off-peak periods. In addition, run the utilities Usage Summary and Usage Deletion during off-peak periods.

LOGGED

Parameter **LOGGED** (as defined in HSISCUST) when set to N, can be used only in version 9 (new function mode), or later. After implementing the product, changes to parameters in DB2 objects can be made with the **ALTER TABLESPACE** or **ALTER INDEX DB2** commands.

Buffer pools

By allocating the appropriate buffer pool to the respective table spaces and indexes, as defined in HSISCUST, you can manage your system resources accordingly. For DB2 performance, first investigate the buffer pools. Check with your site specialist on the types and size of buffer pools that are defined for this product.

Space allocation and utilization

In terms of space utilization, "-1" has been set for all SECQTY to enforce "Sliding Secondary Extents". This enables DB2 to manage secondary extents efficiently, and minimizes extension failures. You need to extrapolate the

PRIQTY for the table spaces and indexes for the large tables according to your requirements. Definitions for these DB2 objects are listed in the respective jobs in JCLLIB.

Repository tables with the biggest impact on performance due to size are TMODULE, TUSEMTD, and TJOBDATA. Data for the TMODULE table is populated during Inquisitor Import process. TUSEMTD, and TJOBDATA tables are populated during Usage Import. For example, you might have more than 300 million usage records in the TUSEMTD table, and more than 20 million modules identified in the TMODULE table. To minimize space utilization and improve SQL query performance, you must prune your usage records by running the Usage Deletion job HSISUDEL.

Declared Global Temporary tables

Declared Global Temporary tables are used in the Match Engine job, and also during the Asset Aggregator process. The TEMP table spaces must be large enough to handle these two processes. In DB2 Version 8, the DSNDB07 work file database does not have 8 K page size defined as part of the DB2 installation. If you are running version 8, you need to define at least one 8 K table space in a TEMP database.

Work file database

When you run some of the SQL queries, they can produce a large amount of output. In order to avoid any excessive output, increase the number and size of the table spaces in the work file database.

Reorganization and RUNSTATS

It is important to run reorganization of the Repository table spaces periodically, especially after Inquisitor Imports, Usage Imports, and Usage Deletion After reorganization of the Repository table spaces, it is also a good idea to run RUNSTATS for these table spaces.

Appendix E. Reporting with Tivoli Common Reporting

Appendix E shows how the Tivoli Asset Discovery for z/OS report package is loaded into the Tivoli Common Reporting product, connectivity to the Repository by way of the database, and how it is used to produce reports.

Note: The supplied Cognos based report package only contains a Cognos Stage 1 model template. It does not contain any pre-configured reports.

The information in this appendix contains screen captures. You might experience some variation if your choice of browser and the level of Tivoli Common Reporting differ. The levels used in this section are:

- Microsoft Internet Explorer Version: 8.0.6001.18702
- Tivoli Common Reporting Version 2.1
- Tivoli Integrated Portal Build Number cf111021.10
- Microsoft Windows XP Professional Version 2002 Service Pack 3

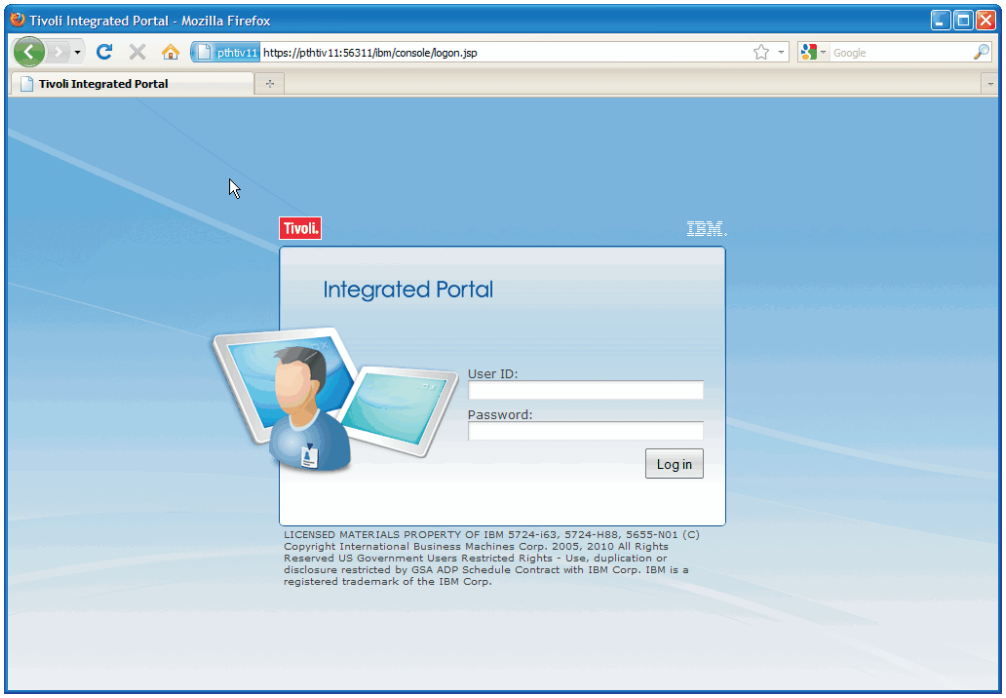
For instructions on how to use Tivoli Common Reporting, see the *Tivoli Common Reporting User Guide SC14-7613* This publication also covers the use of other operating systems.

Following are a number of screen captures which have been reproduced in order to help you understand what is required. Most are portions of the full screen.

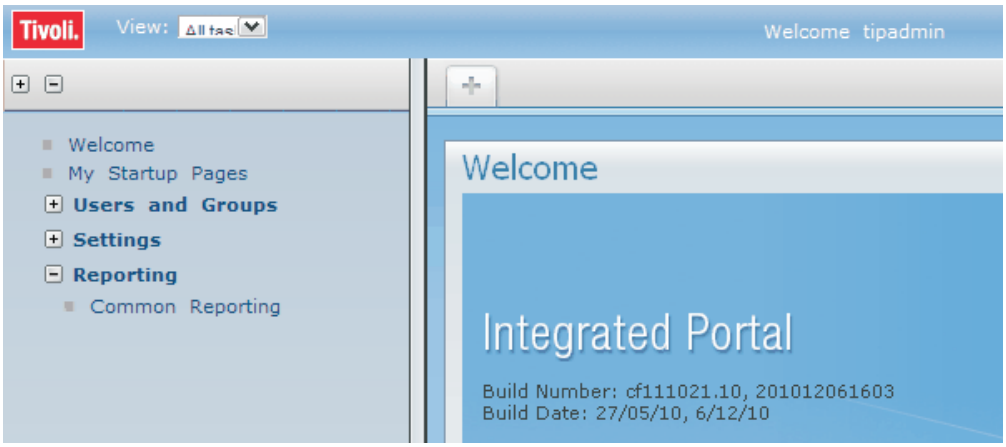
The steps are designed to help you load the Tivoli Asset Discovery for z/OS report package, connect to the Repository, and produce reports.

Phase 1: Load report package for Tivoli Asset Discovery for z/OS into Tivoli Common Reporting

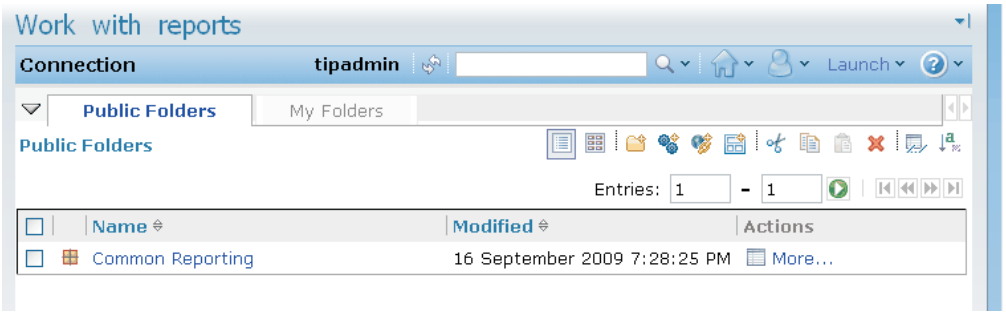
1. Launch the Tivoli Common Reporting product with the browser through the Integrated Portal. You are prompted to enter a user ID and password. The user ID and password are the values created when Tivoli Common Reporting was installed. The launch URL is the one indicated by the Tivoli Common Reporting product when it is installed.



- Enter user ID and password.
- Click **Log in**.
- If valid, a Welcome screen is displayed.

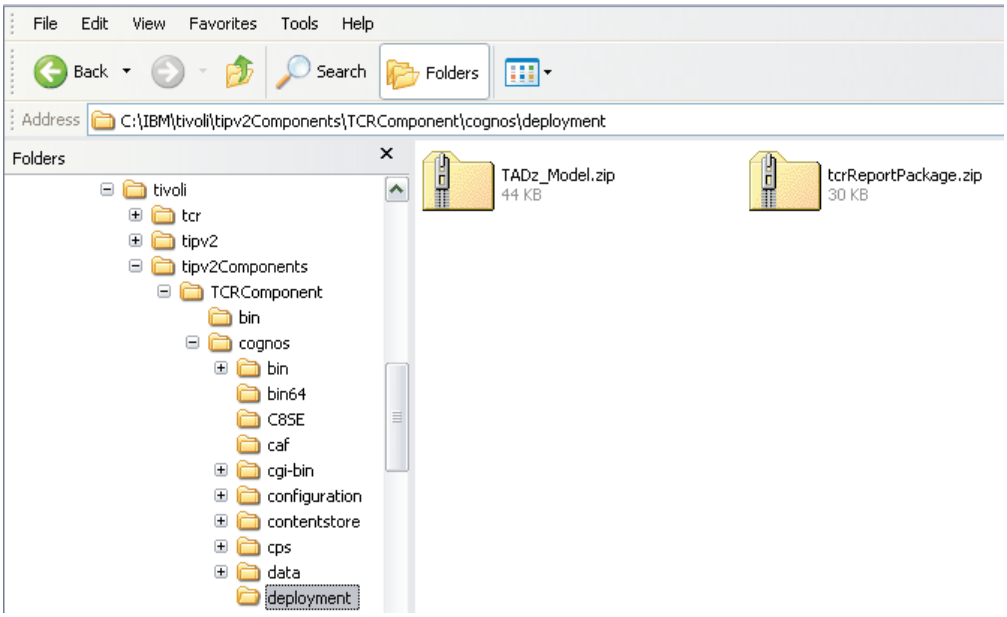


2. Move to the **Work with reports** window.
   - Click **Reporting** followed by **Common Reporting**
   - The **Work with reports** screen is displayed.



If you have not added report packages from other Tivoli products, there is only one entry shown under the **Public Folders** icon.

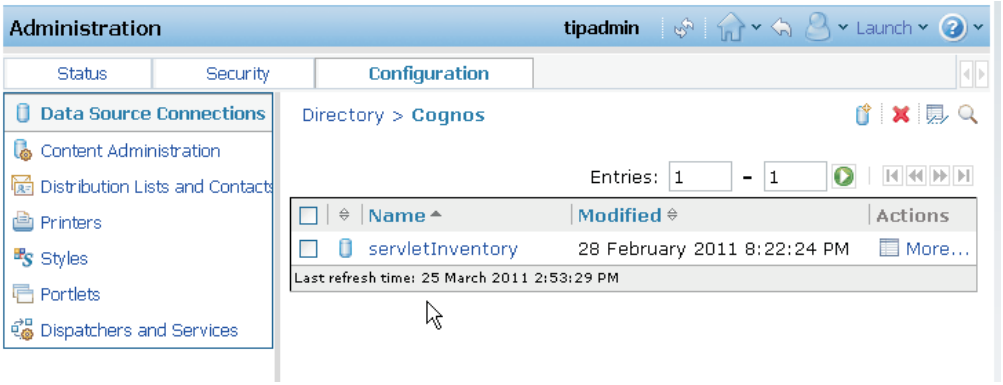
3. You must now locate the report package supplied by Tivoli Asset Discovery for z/OS in `C:\IBM\tivoli\tipv2Components\TCRComponent\cognos\deployment`, and you should find the file `TADz_Model.zip`.



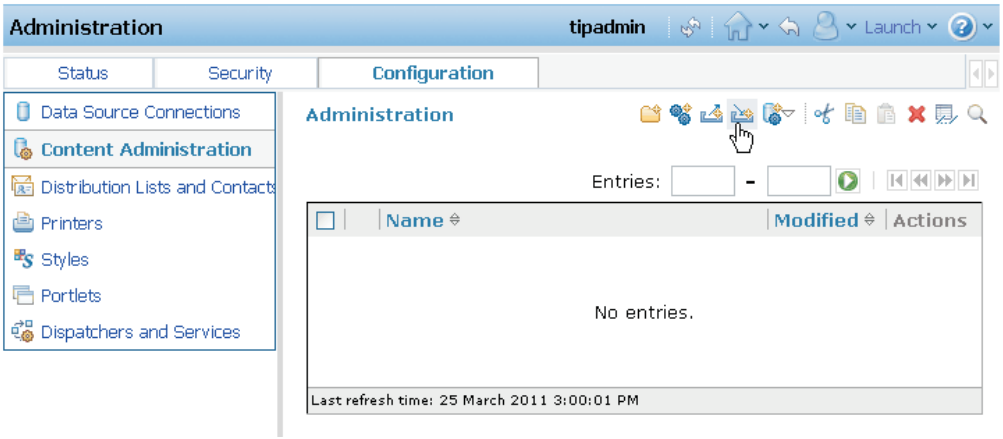
4. To load this report package, you must start the **Administration** component of Tivoli Common Reporting.
   - Select the **Launch** arrow and choose **Administration** from the dropdown menu.

Click the **Configuration** tab.

The configuration options for the **Administration** components that can be modified are displayed.

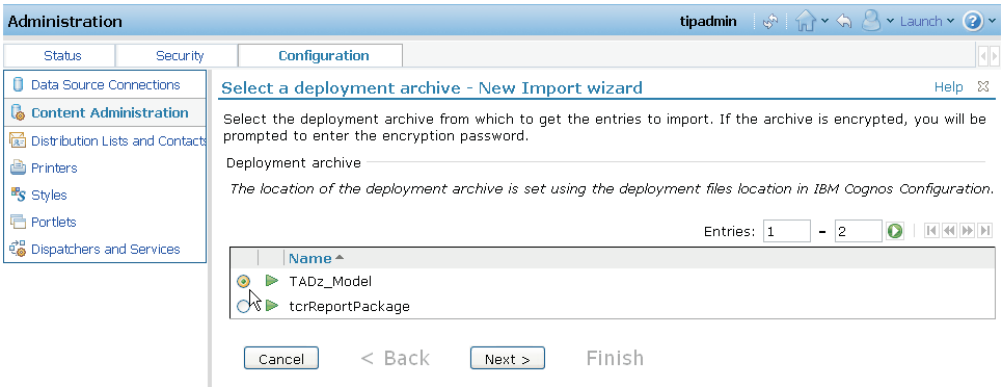


5. To load the report package, click **Content Administration**.

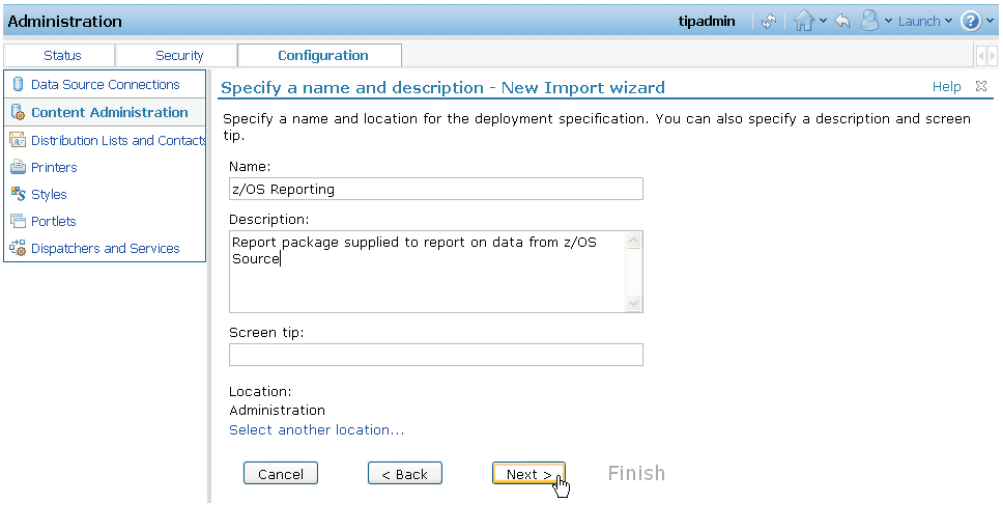


Click the **New Import** icon to open the deployment archive from where you can select the package to import.

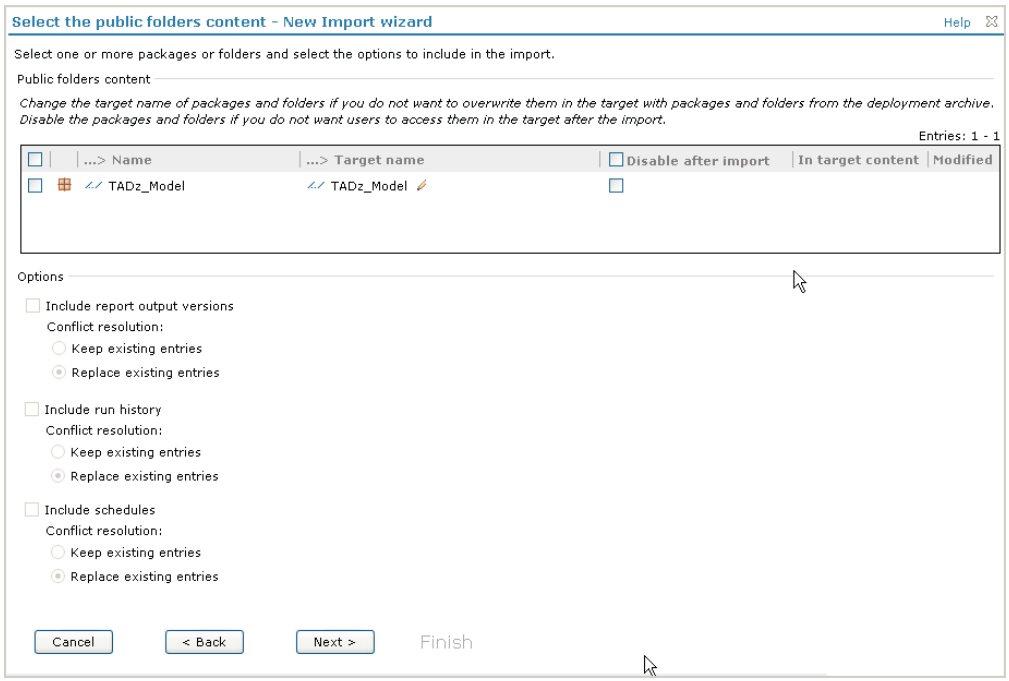
Depending on your screen setting, you might be interrupted by a screen requesting a user ID and password. The user ID and password values are the same as the user ID and password used when entering the Tivoli Common Reporting product.



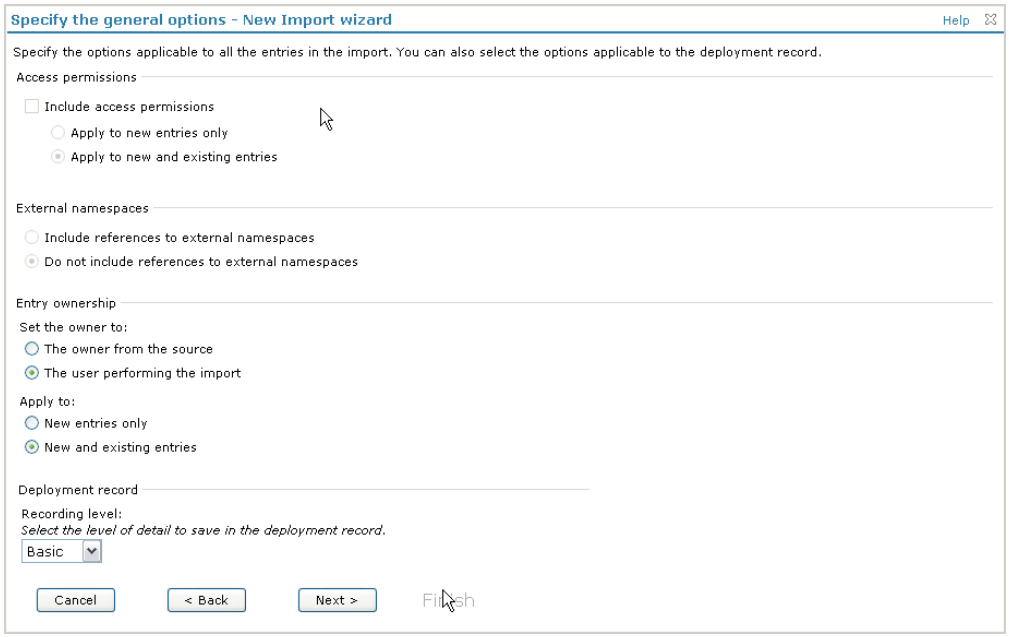
6. From **Select a deployment archive - New Import wizard** pane, select the `TADz_Model` entry and click **Next**.



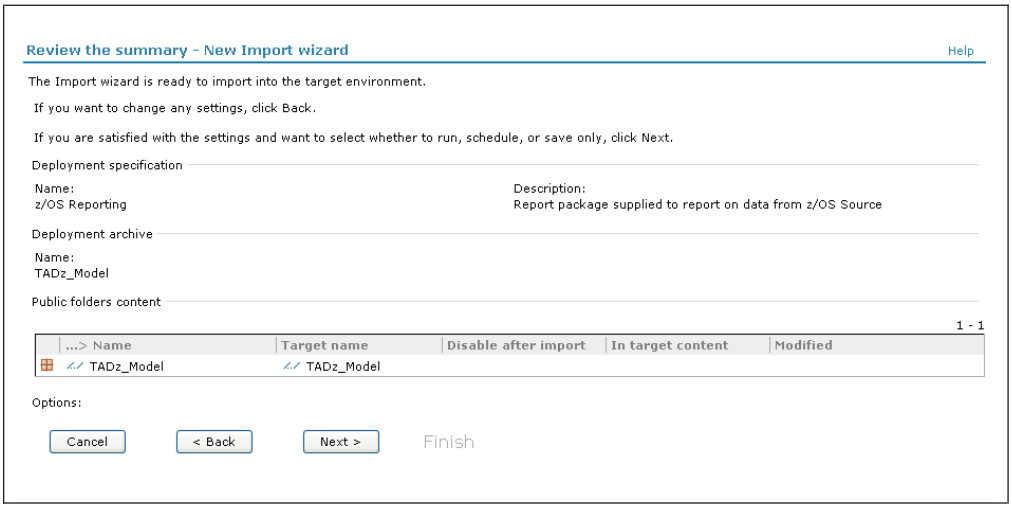
7. On the **Specify a name and description - New Import wizard** pane, enter values:
   - Enter **Name:** In the example shown it is `z/OS Reporting`.
   - Enter **Description:** In the example shown, `Report package supplied to report on data from z/OS Source` is used.
   - In the example the **Screen tip** value was left blank.
   - Click **Next**.



8. On the **Select the public folders content - New Import wizard**
   - Click the box next the name `/../TADz_Model`. This produces the report package you want to use.
   - In the example shown the default option values are used.
   - Scroll to the bottom and click **Next**.



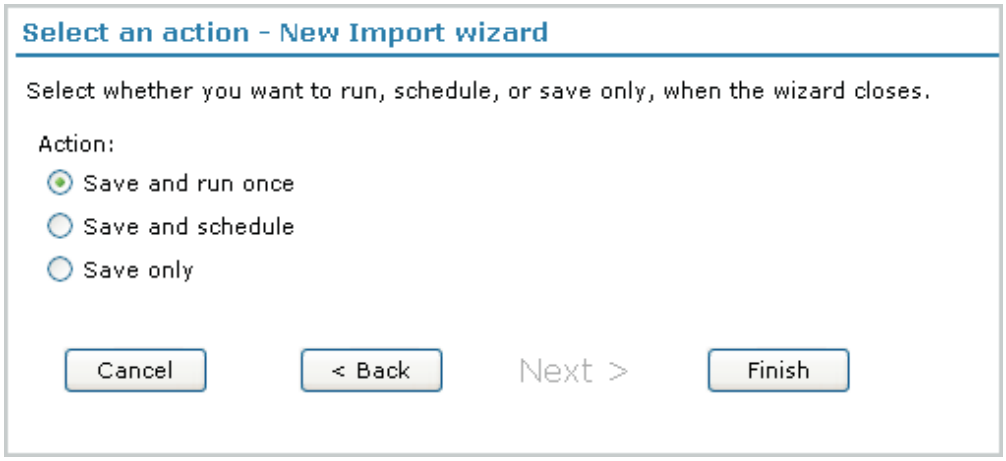
9. On the **Specify the general options - New Import wizard** pane, use the default values given, unless you want to use other options.
   - Scroll to the bottom of the pane and click **Next**.



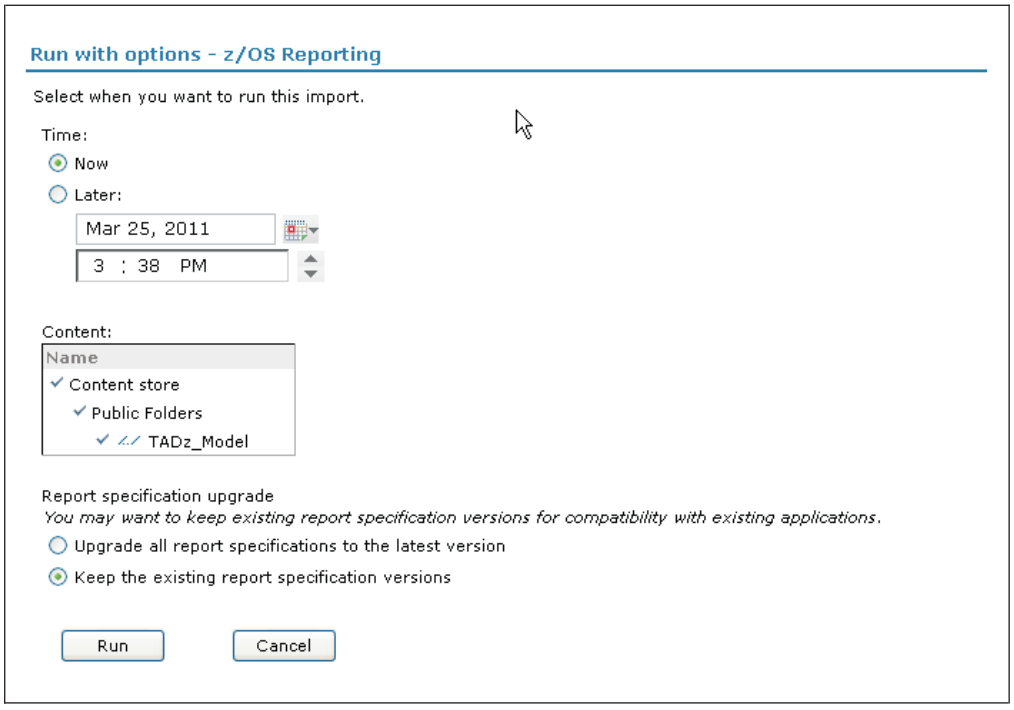
10. The **Review the summary - New Import wizard** screen is the final screen to review your entries. You can use the **Back** button to revise entries.

    As this is the first time you are doing this import, the **Modified** column is shown as blank.

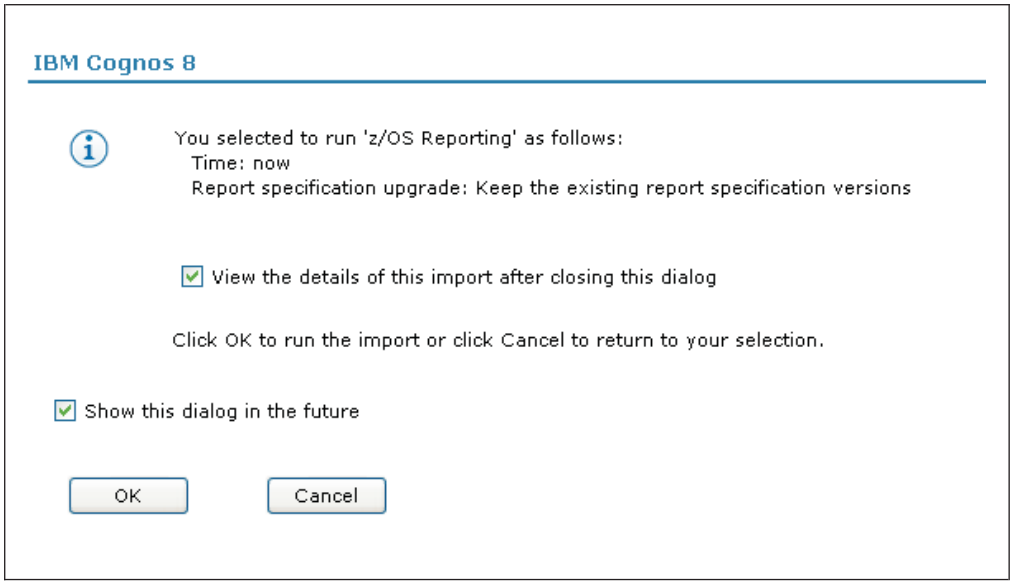
    Scroll to the bottom of the screen and click **Next**.



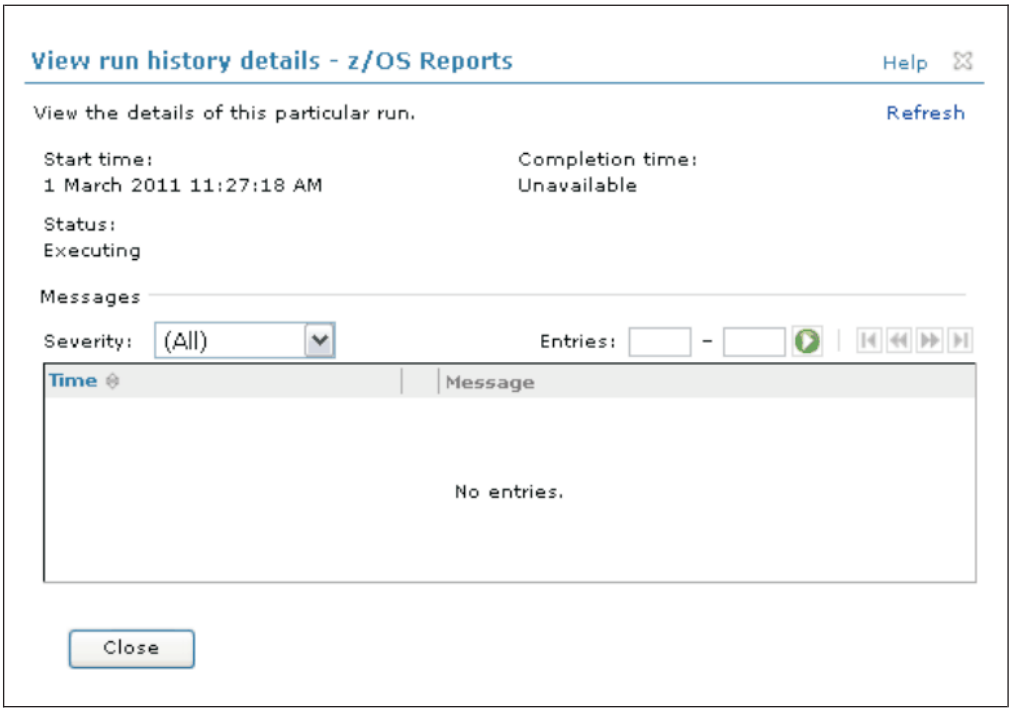
11. On the **Select and action - New Import wizard** pane:
    - Ensure the **Save and run once** action is selected.
    - Click **Finish**.



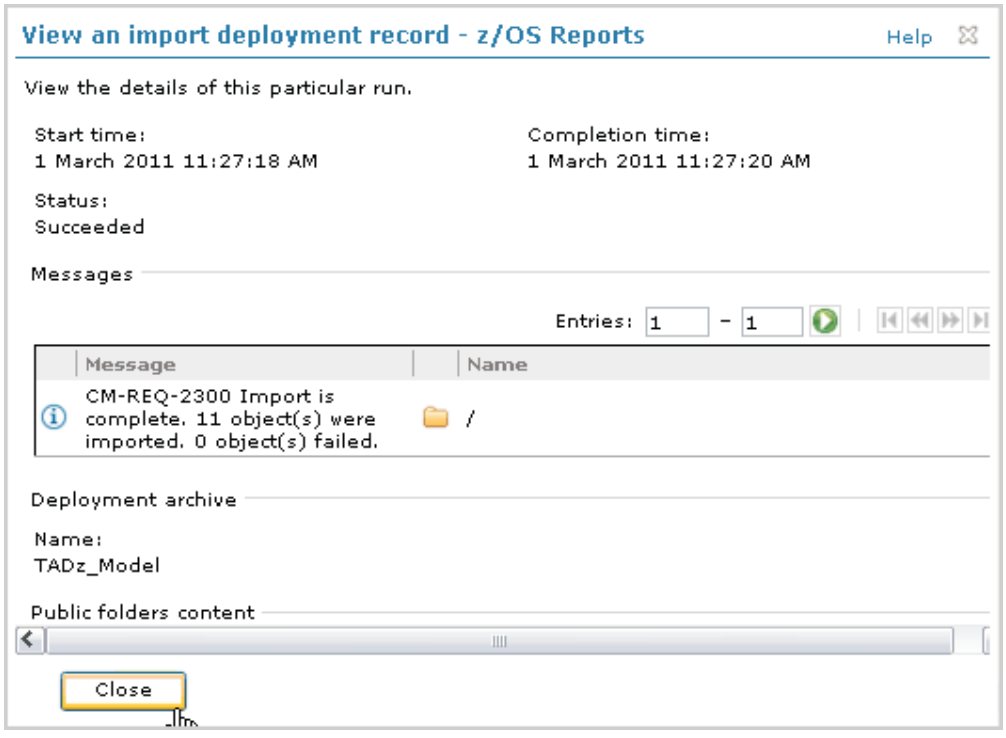
12. On the **Run with options - z/OS Reports** pane:
    - Use the default values to run immediately and retain the existing report specifications.
    - Click **Run**.



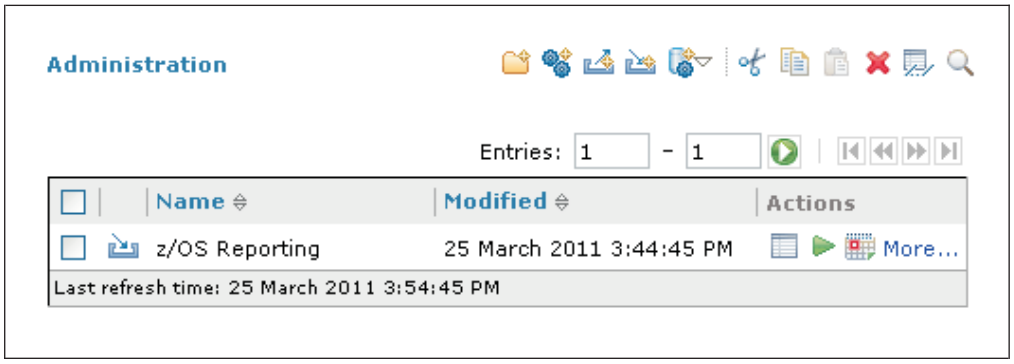
13. The **IBM Cognos 8** screen requests confirmation of your action.
    - Tick box for **View the details of this import after closing this dialog**
    - Tick the box for **Show this dialog in the future**.
    - Click **OK** .



14. The **View run history details - z/OS Reports** screen displays the status of the import.
    - To refresh this screen, click the **Refresh** icon.
    - When completed, the status shows Succeeded.
    - Click **Close**.



15. The administration screen is displayed. This step completes the import of the report package.



Phase 2: Connect with the ODBC client and access the Tivoli Asset Discovery for z/OS Repository

As Tivoli Asset Discovery for z/OS information is kept in a DB2 system on z/OS, an ODBC connection to that database is needed. It might be necessary to obtain the DB2 Connect product, or a DB2 product that has DB2 Connect. Consult with your DBA on connectivity to the Tivoli Asset Discovery for z/OS Repository database. You must consider:

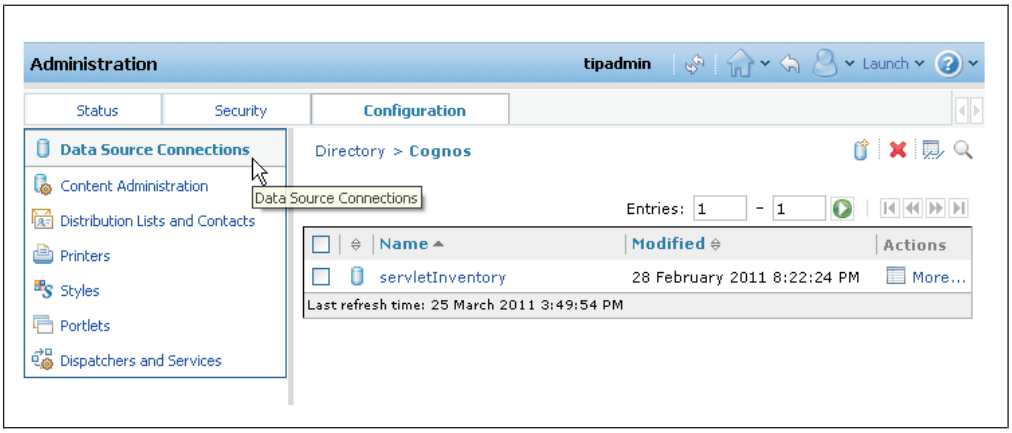
- Installing a product with DB2 Connect capabilities. Refer to step 2 in "(Optional) Installing Tivoli Common Reporting" on page 85.
- Ensure the following post-installation jobs have been run. Refer to Table 6 on page 19 and Table 8 on page 19.
  - HSISGRNT
  - HSISCOGA
  - HSISCOGT
- Performing a bind to load the DB2 packages onto the DB2 subsystem on the host. This requires a minimum authority of BINDADD. This task is usually performed by a DBA. An example of running a bind using a command line is shown here:

  ```
  db2 bind path@ddcsmvs.lst blocking all
          sqlerror continue messages ddcsmvs.msg grant public
  ```

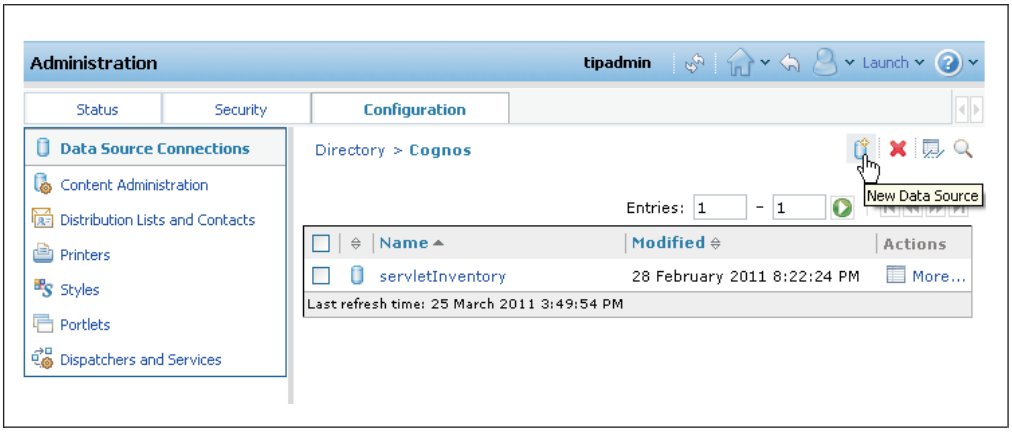
  where path = location of bind list file (for example in Windows it would be: C:\sqllib\bnd) and `ddcsmvs.lst = bind list file`

Once you have established the ODBC connection on the system, you must follow these steps:

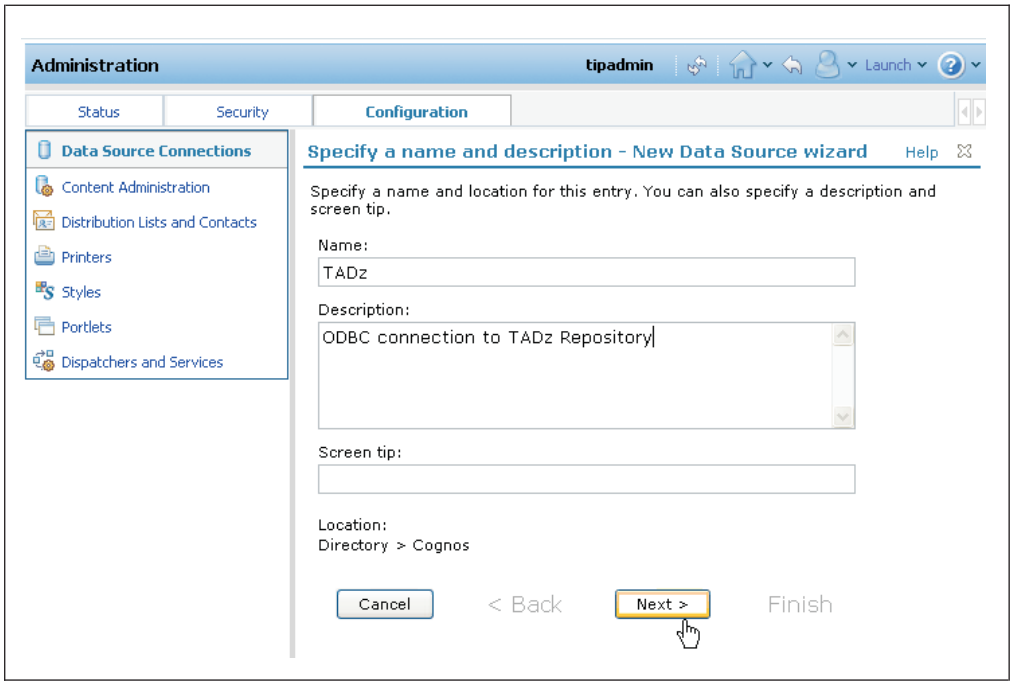
1. Return to the **Administration** screen. This screen is displayed in the last step of Phase 1



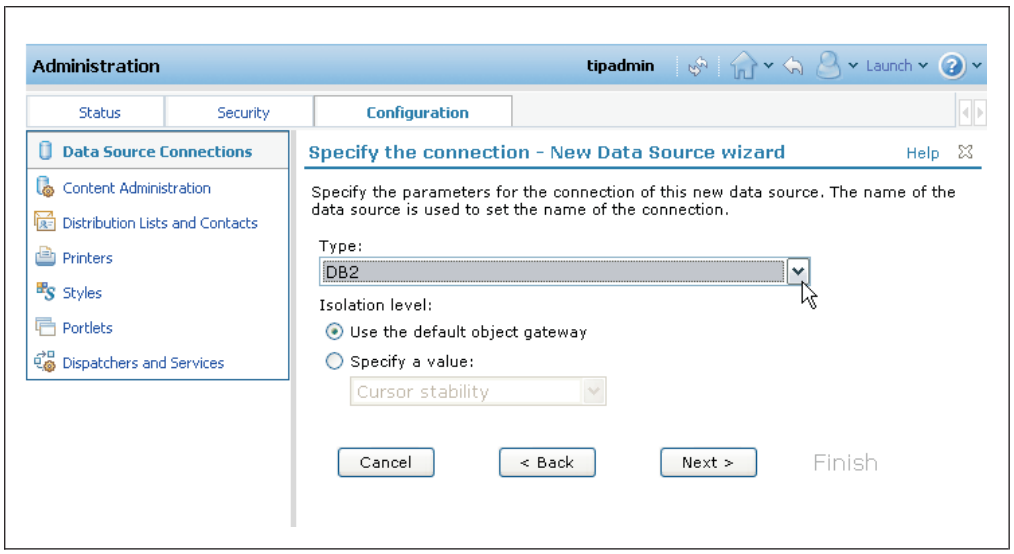
2. Click the **Data Source Connections** icon on the left panel. If your session has timed out you can repeat steps 1-6 from Phase 1.



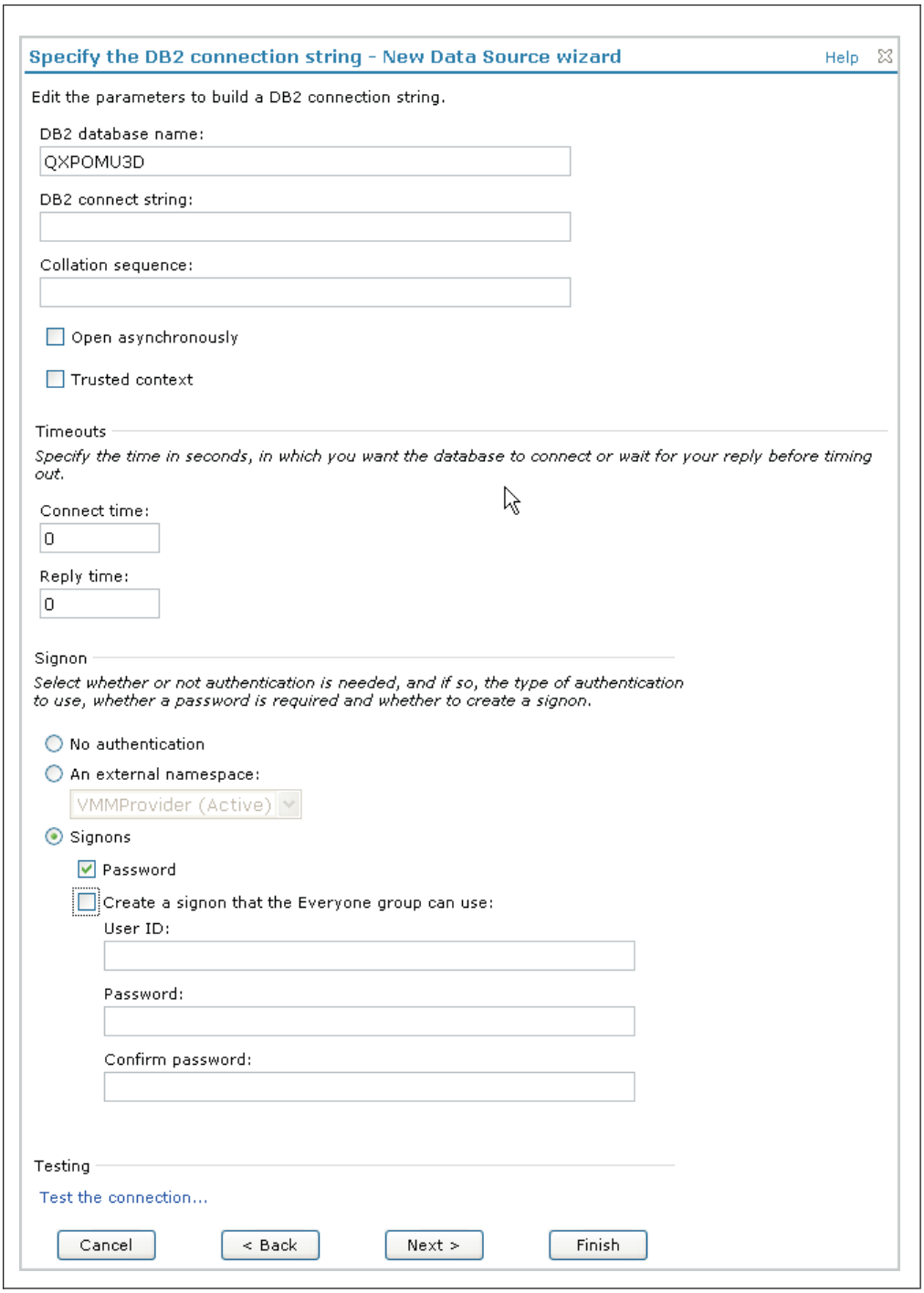
3. You must create a new Data Source Connection. To do this click **New Data Source** from the icon at the upper right of the screen.



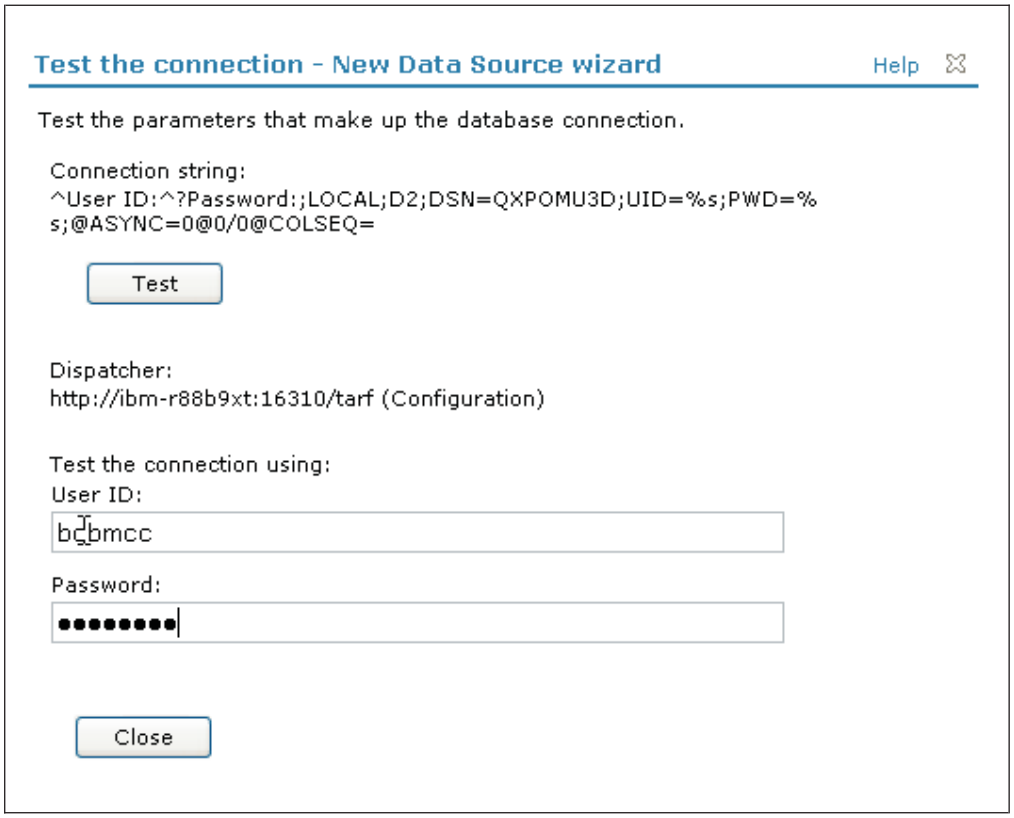
4. On the **Specify a name and description - New Data wizard**
   - Enter a **Name:**. In the example `TADz` is used.
   - Enter a **Description**: In the example `ODBC connection to TADz Repository` is used.
   - Leave the **Screen tip:** field blank.
   - Click **Next**.



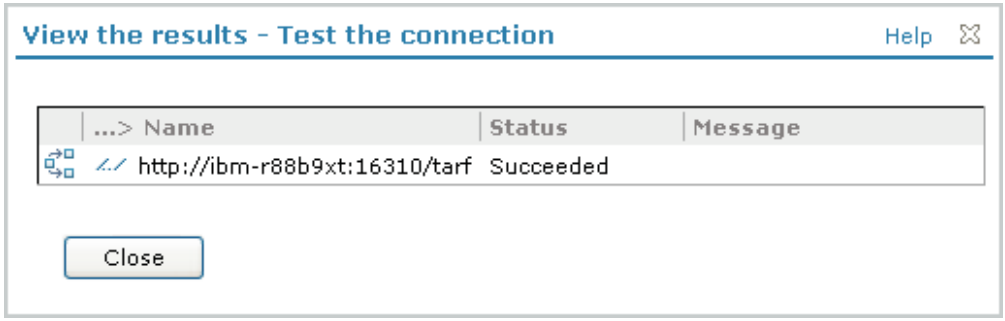
5. On the **Specify the connection - New Data Source wizard**
   - Enter a **Type** value. In the example shown it is `DB2`. This value can also be assigned by the drop-down menu.
   - Under the icon **Isolation level** the value `Use the default object gateway` has been selected. This is the default.
   - Click **Next**.



6. On the **Specify the DB2 connection string - New Data Source wizard:**
   - Enter a **DB2 database name** value. In the example the value is `qxpomu2de92`.
   - Scroll down to the **Password** field, leaving the prior fields blank (unless you want to edit them).
   - Tick the **Password** box.
   - To check these values click the string **Test the connection**.

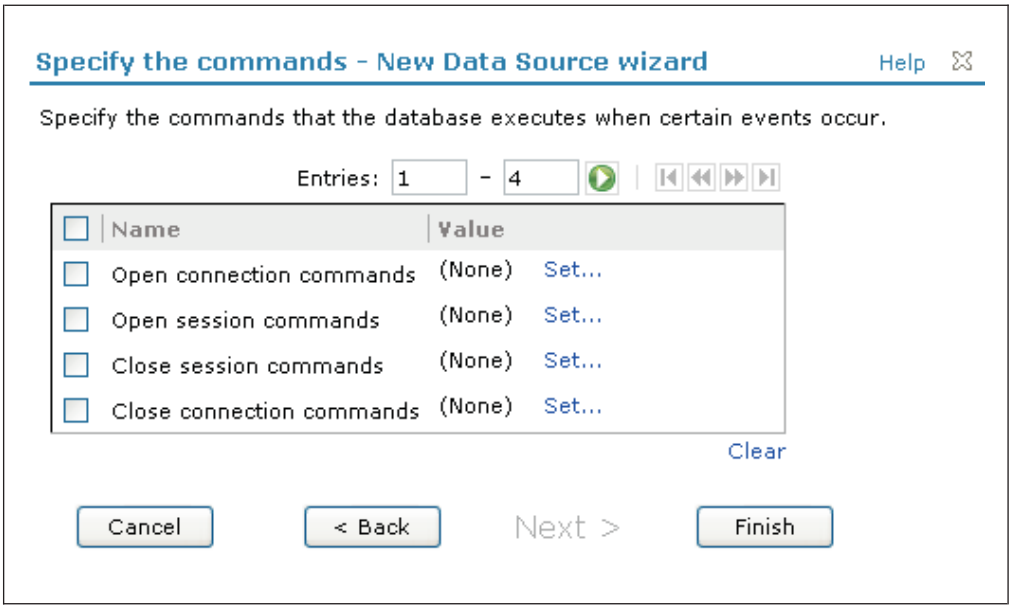


- Click **Test** and, if successful, the results are displayed on the **View the results - Test the connection** pane shown here.

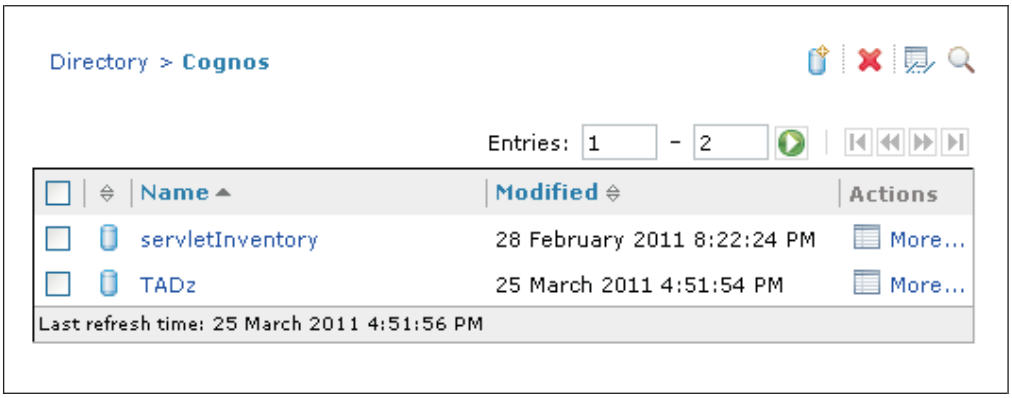


- Click **Close** to return to the **Test the connection - New Data Source wizard** screen.
- From there, click **Close** again, and it returns you to the **Specify the DB2 connection string - New Data Source wizard** screen.

7. On the **Specify the DB2 Connection string - New Data Source wizard** screen, click **Next** .



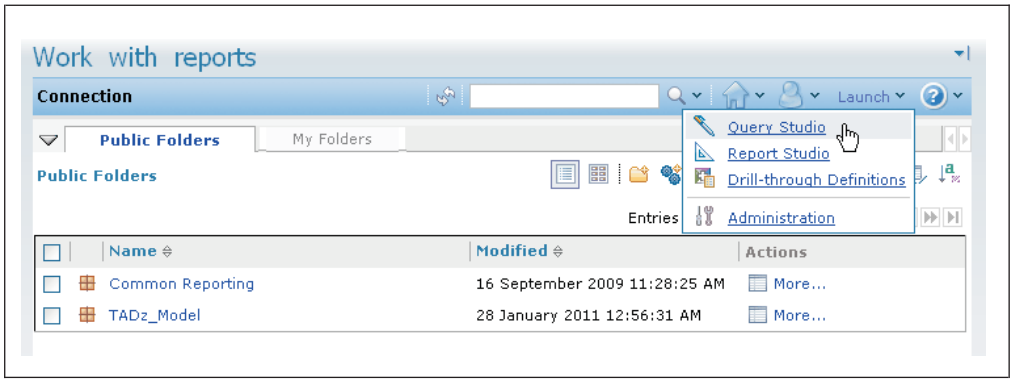
8. On the **Specify the commands - New Data Source wizard** screen, there are no changes to be made. Click **Finish**.



9. The **Directory > Cognos** screen is displayed, showing the available data sources, one of which is `TADz`.
10. You now have an ODBC connection to the Tivoli Asset Discovery for z/OS Repository, as well as the Data Source in Tivoli Common Reporting.

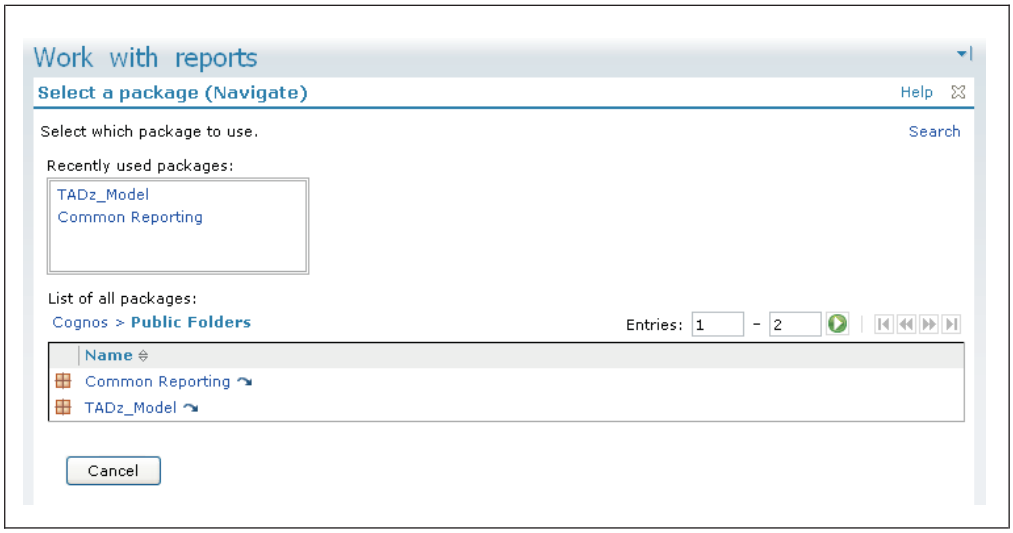
Phase 3: Produce reports

1. Move to the **Work with reports** screen.
   - Click **Reporting** on the left pane.
   - Click **Common Reporting**.

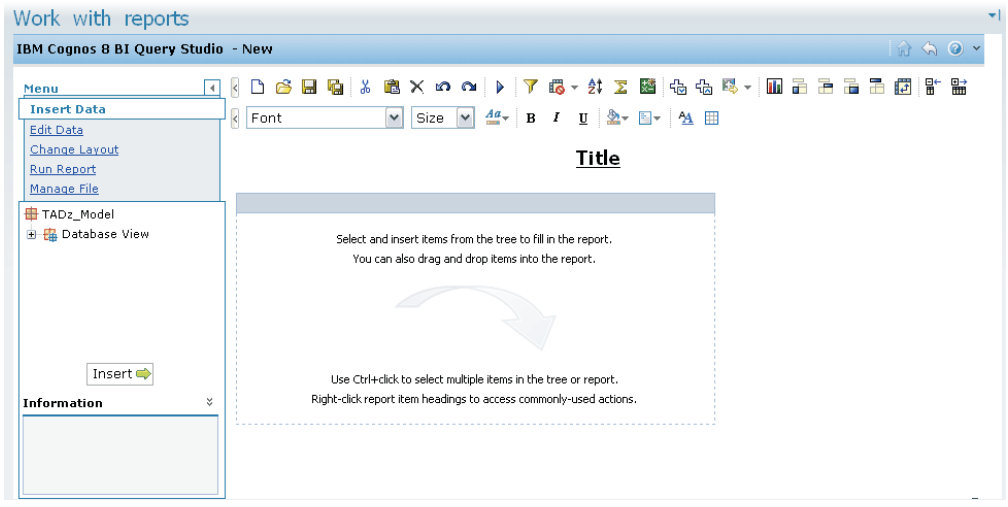


The **Work with reports** screen is displayed.

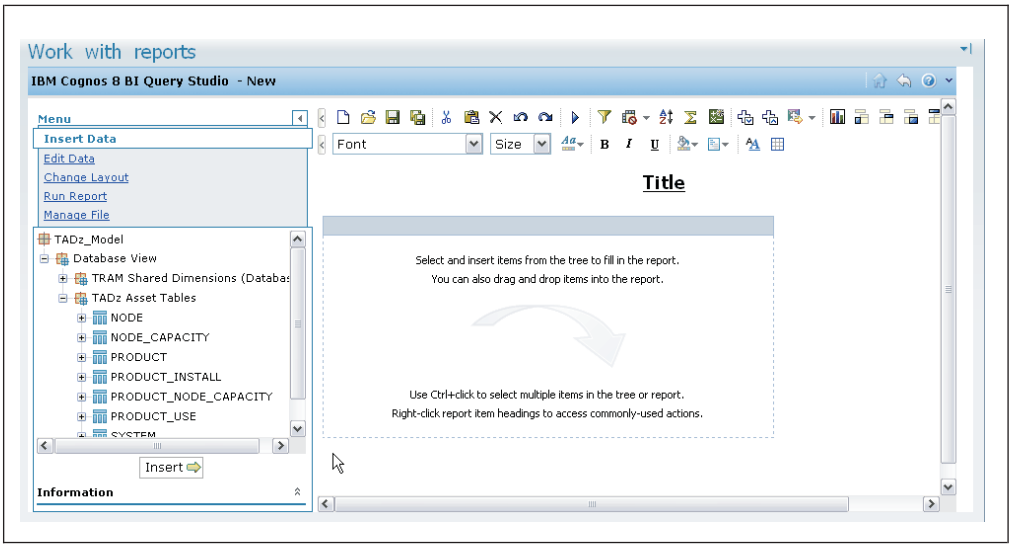
2. To create a report select **Query Studio** from the **Launch** dropdown menu on the right side.



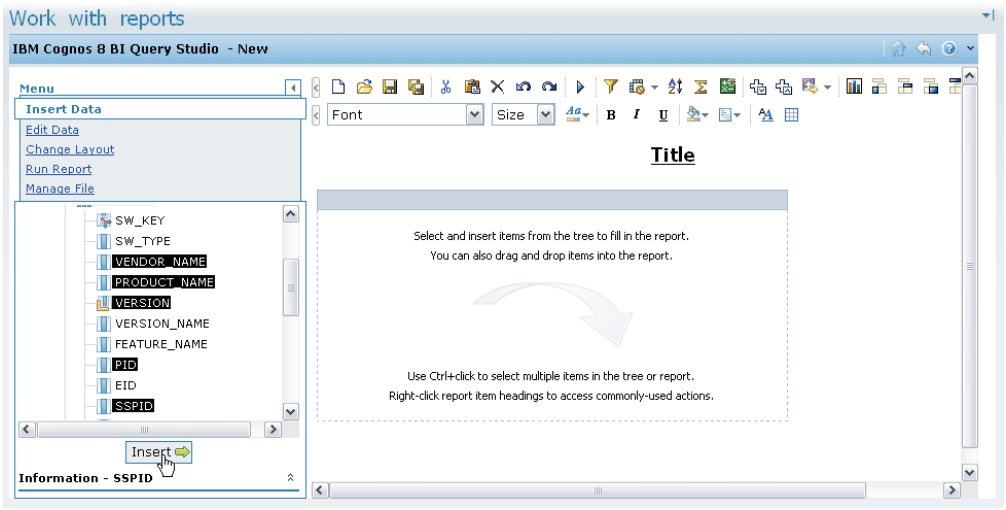
3. From the **Select a package (Navigate)** screen, select the **TAD_z Model** icon. The next panel might take some time to display



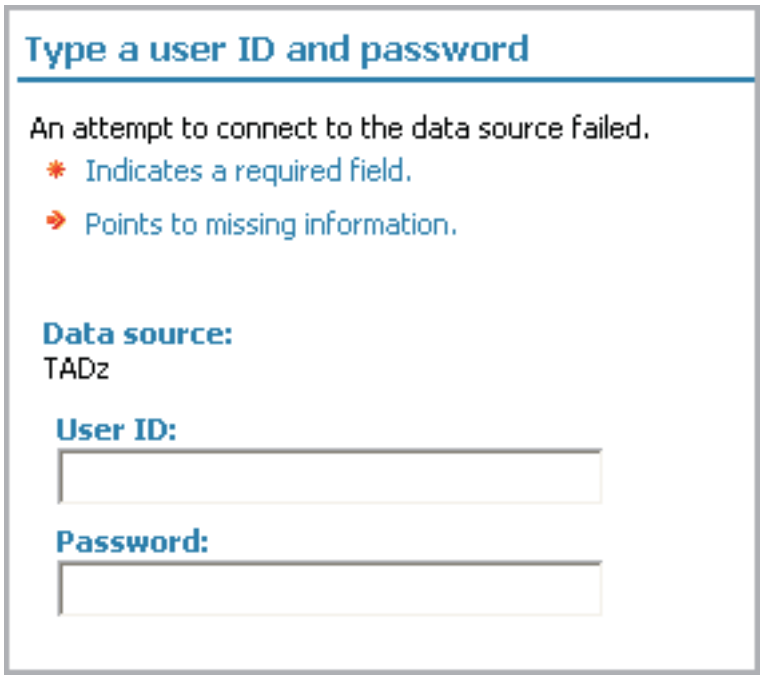
4. The **IBM Cognos 8 BI Query Studio - New** screen is displayed. Click the **Database View** icon.



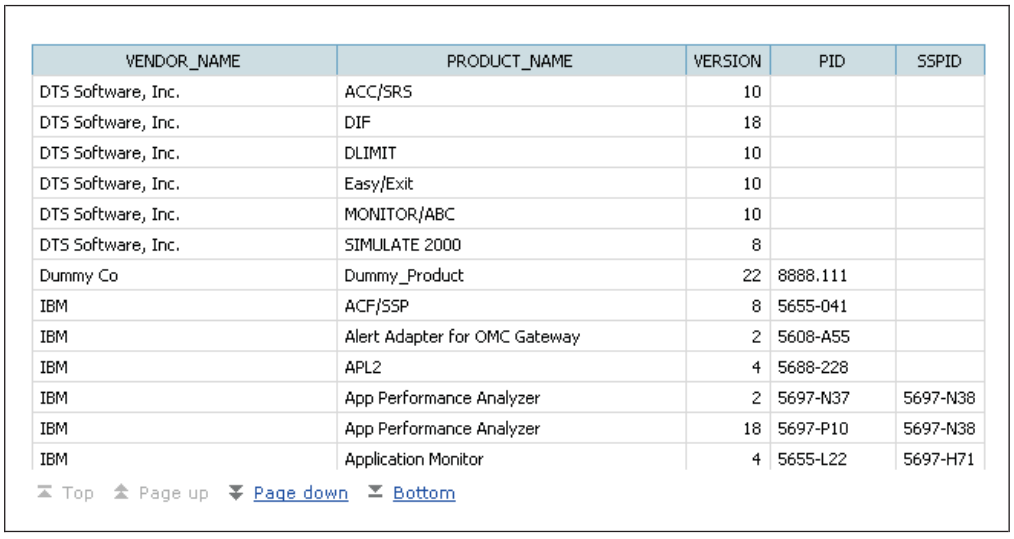
5. Click the + symbol beside **Database View** and then **TADz Asset Tables**. This action expands the fields. It shows all items at the table level for the Tivoli Asset Discovery for z/OS Asset Tables.



6. Click on the field for the table **PRODUCT** to expand it. In order to select multiple fields, hold down the **CTRL** key and mouse click to select fields you want to report on. You can scroll down to other tables but the **CTRL** key must remain held otherwise prior selections are lost. In this example 5 fields on the leftmost pane are selected.
7. In order to report on fields that have been selected scroll to the bottom of the screen and click **Insert**.

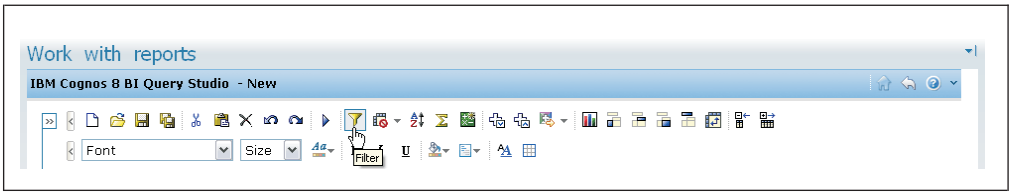


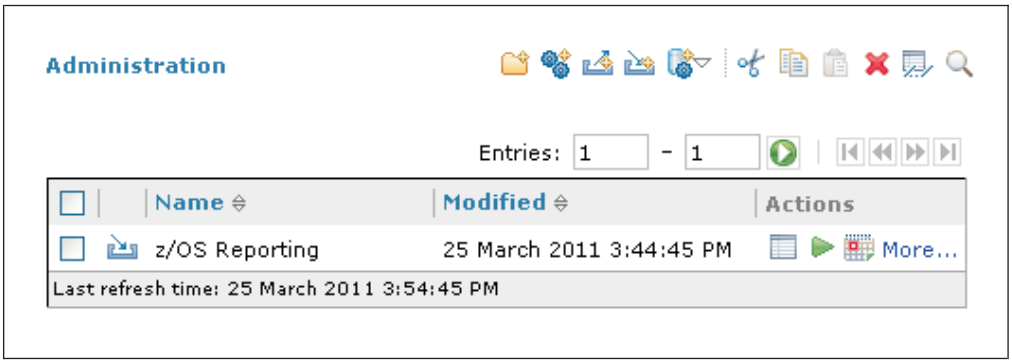
8. Depending on your Data Source Connections setup you might be presented with a panel to supply your TSO user ID: and Password. Populate these fields and click **OK**.
9. The report might take a while to display. When it displays you can use the Top/Page up/Page down/Bottom buttons to navigate through the report.



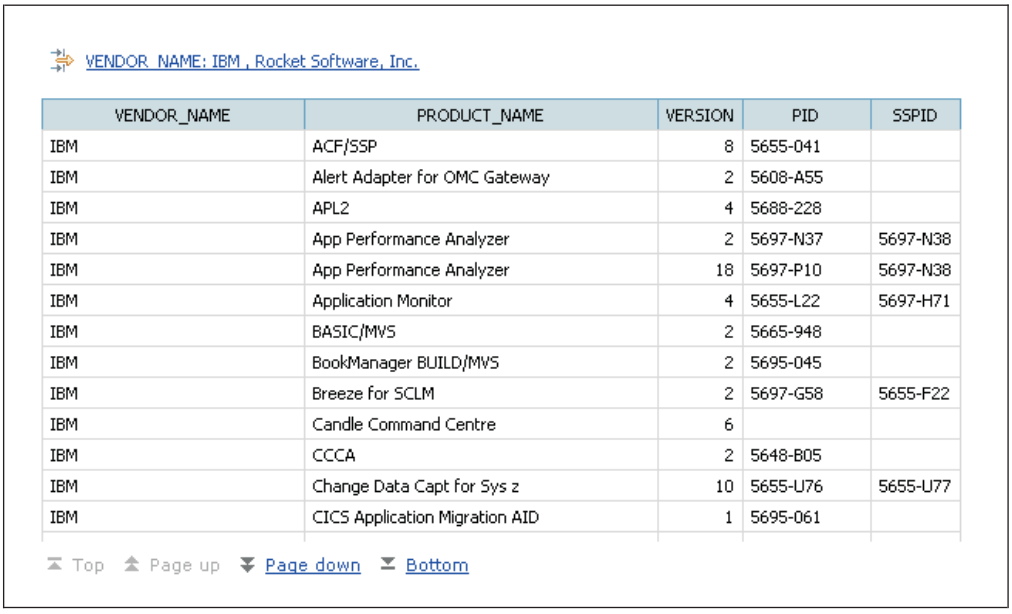
VENDOR_NAME	PRODUCT_NAME	VERSION	PID	SSPID
DTS Software, Inc.	ACC/SRS	10		
DTS Software, Inc.	DIF	18		
DTS Software, Inc.	DLIMIT	10		
DTS Software, Inc.	Easy/Exit	10		
DTS Software, Inc.	MONITOR/ABC	10		
DTS Software, Inc.	SIMULATE 2000	8		
Dummy Co	Dummy_Product	22	8888.111	
IBM	ACF/SSP	8	5655-041	
IBM	Alert Adapter for OMC Gateway	2	5608-A55	
IBM	APL2	4	5688-228	
IBM	App Performance Analyzer	2	5697-N37	5697-N38
IBM	App Performance Analyzer	18	5697-P10	5697-N38
IBM	Application Monitor	4	5655-L22	5697-H71

10. Now you can use the filter and sort actions to fine-tune the report produced. The next graphic shows the filter icon. Do not click on it yet.



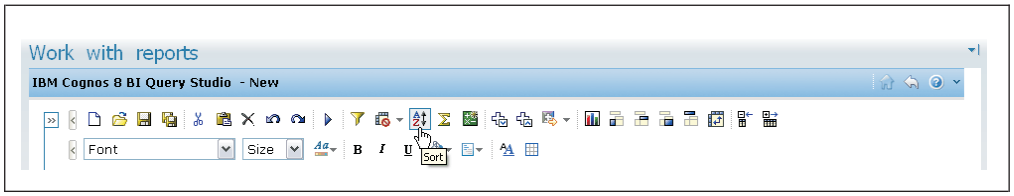


11. As an example, you want to report on any products from IBM and other vendors. Click**VENDOR_NAME**. Then click the **Filter** icon. You now have a **Filter (Pick values from a list)** screen. In this example, the box to the left of **IBM** and **Rocket Software Inc**. have been selected. Also, the box **Prompt every time the report runs**, and **Apply the filter to individual values in the data source** are selected. Click **OK**.



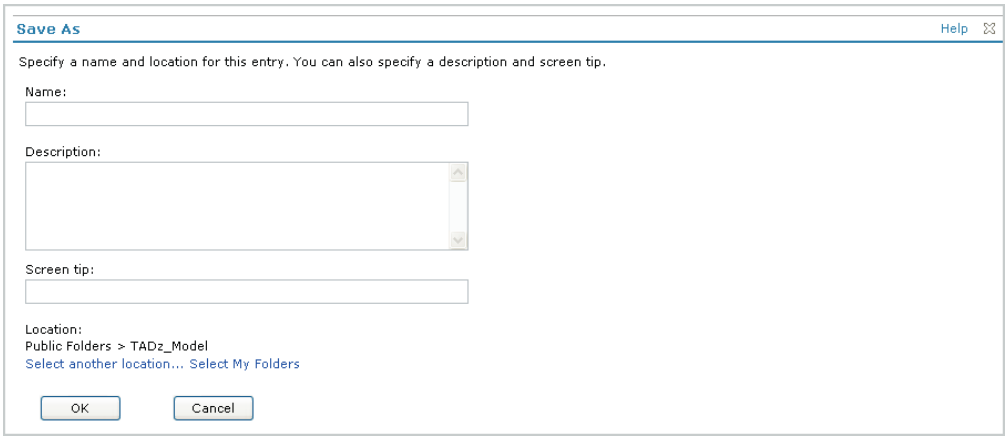
VENDOR_NAME	PRODUCT_NAME	VERSION	PID	SSPID
IBM	ACF/SSP	8	5655-041	
IBM	Alert Adapter for OMC Gateway	2	5608-A55	
IBM	APL2	4	5688-228	
IBM	App Performance Analyzer	2	5697-N37	5697-N38
IBM	App Performance Analyzer	18	5697-P10	5697-N38
IBM	Application Monitor	4	5655-L22	5697-H71
IBM	BASIC/MVS	2	5665-948	
IBM	BookManager BUILD/MVS	2	5695-045	
IBM	Breeze for SCLM	2	5697-G58	5655-F22
IBM	Candle Command Centre	6		
IBM	CCCA	2	5648-B05	
IBM	Change Data Capt for Sys z	10	5655-U76	5655-U77
IBM	CICS Application Migration AID	1	5695-061	

12. The filtered report displays, with a list of Vendor Names noted on top of each page of the report.
13. Use the **Sort** function shown here. Do not click on it yet.



14. On the current report if you click **PID**, and then click the **Sort** icon, you have a report sorted in PID order. **PID** stands for Program Identification.
15. To save a report, use the **Save** icon shown here. Do not click on it yet

16. Click the **Save** icon, and enter **Name:** and **Description:** followed by the **OK** button



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.

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.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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 might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes are incorporated in new editions of the publication. IBM may make improvements and 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 Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements are the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

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 www.ibm.com/legal/copytrade.shtml.

Intel is a registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft and Windows 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.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Related publications

Publications referenced in this guide are listed here:

- *DB2 Connect Quick Beginnings for DB2 Connect Personal Edition, GC23-5839-03*
- *Language Environment Programming Reference, SA22–7562*
- *Language Environment Programming Guide, SA22–7561*
- *Language Environment Customization, SA22-7564.*
- *MVS Initialization and Tuning Reference, SA22-7592*
- *OS/390 V2R10.0 MVS Programming: Assembler Services Guide, (GC28-1762-09)*
- *Program Directory, GI11-8965*
- *Tivoli Common Reporting User Guide , SC14-7613-00*

Supplied on DVD

- *IBM Tivoli Common Reporting (LCD7 - 1995–02)*

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.

The major accessibility features in this product enable users to do the following:

- Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on the screen.

Magnifying what is displayed on the screen

You can enlarge information on the product windows using facilities provided by the operating systems on which the product is run. For example, in a Microsoft Windows environment, you can lower the resolution of the screen to enlarge the font sizes of the text on the screen. Refer to the documentation provided by your operating system for more information.

Index

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Printed in USA

SC27-2701-01