IBM WebSphere Studio Asset Analyzer for Multiplatforms
Version 5 Release 1

Best Practices Guide

SC19-1168-01

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 65.

Second Edition (December 2007)

This edition applies to Version 5 Release 1 of IBM WebSphere Studio Asset Analyzer for Multiplatforms (Program Number 5655-R10).

This edition also applies to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

You can order publications online at www.ibm.com/shop/publications/order, or order by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. Eastern Standard Time (EST). The phone number is (800)879-2755. The fax number is (800)445-9269.

You can find out more about WebSphere Studio Asset Analyzer for Multiplatforms by visiting the IBM Web site for the product at: www.ibm.com/software/awdtools/wsaa/.

A form for reader's comments appears at the back of this publication. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About this document

This guide contains considerations and guidelines for using WebSphere Studio Asset Analyzer. It is not a comprehensive guide but a supplement to the product online help and the other documents for WebSphere Studio Asset Analyzer. Use this guide in conjunction with the other product documentation to gain a better understanding of the product. To find other product documents, visit the Library section of the WebSphere Studio Asset Analyzer product Web site at www.ibm.com/software/awdtools/wsaa/.

You are encouraged to use these best practices wherever possible in deploying and using the product, but temper their application with your own experience and knowledge of your installation environment and users. Most of the practices and information in this edition of the guide apply to MVS™ assets.

This document refers you to the following sources of product documentation:

- RELATED TASK HELPS refers to the product help topics that are accessible from the **Task Help** list on the product Home page.
- RELATED ONLINE HELP refers to the product help topics that you access by clicking the ? icon on any product page.

  References to online help topics are expressed in the hierarchy of the navigation pane, for example, **Product overview > Impact analysis > Impact analysis scope**.

  A searchable copy of the online help is available on the Web, which you can access as part of the IBM® zSeries® Application Development Tools information center at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp.
- RELATED REFERENCES refers to the PDF documents that are available from the PDF documents entry in the online help navigation pane or to other IBM manuals (for products such as CICS, IMS, and DB2) that are available from the IBM Product publications site.
- RELATED TECHNOTE refers to technotes available from the Support page of the WebSphere Studio Asset Analyzer Web site at http://www.ibm.com/software/awdtools/wsaa/support/.

Other sources of information

WebSphere Studio Asset Analyzer provides a plugin for IBM Support Assistant (ISA). By enabling WebSphere Studio Asset Analyzer in ISA, you can more easily search product, support, and education resources and more easily open PMRs. See http://www.ibm.com/software/support/isa/ for information and to download the free ISA software. After you have installed ISA on your workstation, you can access the WebSphere Studio Asset Analyzer plugin from the WebSphere Studio Asset Analyzer Web site, the ISA Web site, or the ISA product user interface.

Visit the IBM Education Assistant web site at http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wsaa/wsaa/wsaa_coverpage.html and learn about WebSphere® Studio Asset Analyzer from the published materials. IBM Education Assistant is an Eclipse-based Help Center that provides a collection of multimedia educational content (presentations, Show Me Demonstrations, tutorials, and resource links) designed to help you use IBM software products more easily and effectively.

Part 1. Introduction

Chapter 1. General best practices

While using WebSphere Studio Asset Analyzer, observe the following general guidelines and practices:

- Practice by using a variety of scenarios to understand how the product functions and responds.
- Review the online help as you go along, confirming that information is available and that it makes sense to you. There is value in knowing the layout of the help and familiarizing yourself with the content. If you find limitations or lack of clarity in the information, let us know so that we can address the issue. You can use the form at the back of this guide.
- Periodically check the WebSphere Studio Asset Analyzer Web site to see if there are updates to the product publications, support technotes and downloads, Redbooks™, or other product news. The Web site provides helpful links that quickly connect to useful services, such as Preventive Service Planning. There is a direct link from the online help: **Web resources > IBM WebSphere Studio Asset Analyzer** and from the task help: **Getting additional information > Accessing the official product page**.
- Participate in the WebSphere Studio Asset Analyzer Newsgroup to learn from the experience of others and to share your experiences. To access the newsgroup, click **Web resources > Newsgroup** in the online help. For help with setting up and using the newsgroup, click **Getting additional information > Using the newsgroup** in the task help.
- After you have configured and deployed the product, consider drafting your own best practices. This can be a checklist or other documentation that can be very helpful to users, especially if your site imposes standards that are specific to WebSphere Studio Asset Analyzer, such as naming conventions, usage practices, and prohibitions. Your own best practices document can be a practical mechanism to record the information for consumption by your user community and can be started with minimum effort, and then updated periodically as your organization gains experience with the product.
- Visit the IBM Education Assistant web site at http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wsaa/wsaa/wsaa_coverpage.html and learn about WebSphere Studio Asset Analyzer from the published materials. IBM Education Assistant is an Eclipse-based Help Center that provides a collection of multimedia educational content (presentations, Show Me Demonstrations, tutorials, and resource links) designed to help you use IBM software products more easily and effectively.

Chapter 2. Determining administrative privileges

WebSphere Studio Asset Analyzer has two classes of users, administrators and non-administrators, and provides the ability to restrict non-administrators to a subset of tasks and actions. Generally speaking, non-administrators are restricted to viewing the assets in the database. The following table lists the tasks that non-administrators can and cannot perform:

Table 1. Task permission for non-administrators

Can perform...	Cannot perform...
• Navigate all explore, summary, and details pages. • Create and update applications. • Create and update asset annotations. • Create and delete their own impact analyses. • Create and copy custom queries; edit their own custom queries. • Explore, traverse, and refresh details for a specific user-defined relationship.	• Take an inventory or any inventory-related activity such as creating or assigning concatenation sets, queuing files or containers for analysis, and configuring COBOL and PL/I scanning options. • View summary information about the state of the database. • Delete assets from the database. • Delete, import, export, validate and clean up expired custom queries. • Create, delete, update user-defined relationships, or explore summary of user-defined relationships.

Actions available for administrators are those that can dramatically change the inventory database, that is, adding and deleting metadata, and involve longer running queries (such as those from the Database status pages). Therefore, most installations prefer to control which users have the ability to perform such tasks. It is important that non-administrators know whom to contact in the event that any of these tasks need to be performed. The following table lists the actions that are limited to user IDs with administrator authority, and hidden from view for non-administrators:

Table 2. Actions limited to administrators

Action
• Assign a concatenation set for a file • Assign a default concatenation set for an application • Assign a default concatenation set for a container • Assign a default concatenation set for a site • Assign a new language to a file • Create a new distributed scan root • Create a new concatenation set • Create a new site • View and act upon Database > Analysis queue information • View and act upon Database > Errors information • View and act upon Database > Inventory information • View and act upon Database > Unreferenced assets • View and act upon Database > Work queue • View and act upon Database > Distributed scan roots • View and act upon Database > Runtime environments • View and act upon Database > JavaProcessor patterns • Delete analysis queue entries • Delete applications, bookmarks, concatenation sets, custom queries, user-defined relationships • Delete impact analyses[1] • Delete MVS assets • Update the User field in the Bookmark wizard • Update the User field in the Custom query wizard • Queue a distributed scan root for deletion • Queue a distributed scan root for scan • Queue ChangeMan Application for analysis • Queue container for analysis • Queue File(s) for analysis • Update a distributed scan root • Update a concatenation set • Update a site definition • Clean up expired custom query • Edit a custom query[2] • Export a custom query • Import a custom query • Validate a custom query • Create a user-defined relationship • Update a user-defined relationship • View summaries of user-defined relationships (using **Explore > User-defined relationships**) • Configure COBOL and PL/I scanning options for files/programs, applications, containers, and sites • View summaries of COBOL and PL/I scanning options (using **Database > COBOL and PL/I scanning options**)

Table 2. Actions limited to administrators (continued)

Action
Notes: 1. Can delete any analysis, regardless of who created it. Non-administrators can delete only their own analyses. 2. Can edit any custom query, regardless of who created it. Non-administrators can edit only their own custom queries.

You establish administrator permissions in two places: during configuration using DMHINSTL, Option 0 and during the ISMP installation, if you are installing the runtime on Windows® or AIX®. If you are installing the runtime on Windows or AIX, you should specify the same administrators that were used in DMHINSTL, Option 0.

You can grant or revoke administrator permissions at any time through DMHINSTL Option 0, Set up, by modifying the **Administrator userids** field in the **Server settings** section. After saving your changes, you must restart the WebSphere Studio Asset Analyzer enterprise application. By default, the **Administrator userids** field is set to "*" indicating that all user IDs have administrative authority for WebSphere Studio Asset Analyzer functions. (If you have a WebSphere Studio Asset Analyzer Windows or AIX server, you must also update the Common.cfg file there and restart that server.)

RELATED TASK HELPS
Getting started > Learning the types of users

RELATED REFERENCES
Chapter 3. Defining product defaults, *Configuration and Migration Guide*

Part 2. Administrative practices

Chapter 3. Taking an inventory

MVS assets

Taking an inventory of MVS assets involves three types of scans. While not outwardly visible to users, these scans are managed and coordinated by the WebSphere Studio Asset Analyzer analysis queue processor. The following three types are employed:

Container scan
Creates a list of files in a container and gathers container attributes metadata

Classification scan
Identifies the language and file type and gathers language-neutral metadata (a shallow scan)

Analysis scan
Gathers language-specific metadata (a deep scan)

Source: You take an inventory of source assets on the following three schedules:

- Initial database load, using the browser Inventory wizard or batch update interface
- On-demand requests, using the browser **Queue for analysis** actions
- Incremental load, using the container **Queue for analysis** action or the batch update interface.

Batch: A batch update interface program (DMH0700) is provided in the *yourHLQ*.CONFIG.CNTL(DMHJCLBU) file. This interface is supplied as an alternative to the Inventory Wizard provided in the browser interface, but it provides only a subset of the Inventory Wizard's functionality. Specifically, it does not provide concatenation set handling and scan option setting, and is limited to two-level application assignment. The use of this interface is aimed at allowing users to quickly submit delete file, scan container, and scan file requests using 80-character input records. The record formats for this interface are documented in the *yourHLQ*.CONFIG.CNTL(DMHJCLBU) file.

RELATED ONLINE HELP
Taking an inventory > Taking an inventory of MVS assets > Scanning source code > Scanning without the Web browser

RELATED TECHNOTE
Running DMH0700 while DMH6000 is also running

CICS®: You take an inventory of CICS assets by using the WebSphere Studio Asset Analyzer ISPF interface.

DB2®: You take an inventory of DB2 catalog assets by using the WebSphere Studio Asset Analyzer batch JCL.

IMS™: You take an inventory of IMS assets by using the WebSphere Studio Asset Analyzer ISPF interface.

Software configuration management (SCM) systems: You can use the batch update interface program *yourHLQ*.SDMHCNTL(DMH0700) to specify Changed and Deleted asset information to WebSphere Studio Asset Analyzer. Because every SCM has a point at which assets are promoted, there is an opportunity to notify WebSphere Studio Asset Analyzer that an asset needs to be rescanned and to schedule the rescan by using a batch job. Taking an inventory of source that is under a source-management system involves additional considerations. Review "Applications with source management systems" on page 34 for additional information.

Distributed assets

Taking an inventory of distributed assets involves scanners and analyzers running on a Windows or AIX server to identify, analyze, and classify both physical components (containers and files) and logical components (classes, methods, and fields). WebSphere Studio Asset Analyzer can search the following enterprise servers for programming resources:

- ClearCase® 2003 versioned object bases
- Windows and AIX file systems
- PVCS 6.8.0.0
- CVS client 1.11.1.2 (Build 41)
- WebSphere Application Server installation

RELATED TASK HELPS

Taking inventory > Scanning resources > Scanning ClearCase, CVS, PVCS repositories
Taking inventory > Scanning resources > Scanning WebSphere Application Server
Taking inventory > Scanning resources > Scanning workstation directories and files

RELATED ONLINE HELP

Taking an inventory of distributed assets > Setting up scanners to take an inventory > Choosing scanners and analyzers

Best approach

MVS assets

The following list gives the best approach for taking an inventory:

1. Review existing production concatenations for the source and applications of interest.
2. Create concatenation sets.
3. Create applications.
4. Identify included source files (copy, include, macro, proc). This step is required only if you are unsure of the concatenation sets that should be used to scan your source files.

   Tip: In general, if you have properly defined your concatenation sets and assigned them to your source containers, applications, or sites, WebSphere Studio Asset Analyzer automatically scans only those included source files that are used by your source. This can result in less data being loaded into the database, because only the referenced included source files are loaded. If you are not certain of the concatenation sets that should be used for scanning your source, you can instead scan all the included source containers and let WebSphere Studio Asset Analyzer automatically determine a concatenation set to use.

   a. Take the inventory of identified included containers.
   b. Review inventory results.

5. Take the inventory for source program and batch job containers, assigning concatenation sets and applications.
6. Review inventory results.
7. Scan CICS assets, if applicable.
8. Review inventory results.
9. Scan DB2 Catalog assets, if applicable.
10. Review inventory results.
11. Scan IMS assets, if applicable.
12. Review inventory results.
13. Review run unit and other postprocessing results.

MVS postprocessor reports

MVS postprocessing generates two data definition (DD) statements, SUMMARY DD and RDPRINT DD, which provide information about processing that occurs during postprocessing. To check the status of the postprocessing job after it completes, log onto TSO and check the held output queue on MVS for error messages.

The SUMMARY DD has basic information about the processing of the queue, and the RDPRINT DD provides the details. In the SUMMARY DD, start and stop messages are printed for each processed queue type. A message indicating the return code and the name of the item is printed for each processed item within that queue type. For example:

```
DMH1761I: Start processing queue type 11. Identify File Type - waiting for
          postprocessing to be invoked.
DMH1762I:   Queue type=11 RC=0 for File PROD.SOURCE(GOODSRC1)
DMH1762I:   Queue type=11 RC=0 for File PROD.SOURCE(GOODSRC2)
DMH1762I:   Queue type=11 RC=8 for File PROD.SOURCE(PROBSRC1)
DMH1763I: Stop processing queue type 11.  Highest RC=8
```

If a processed item has a nonzero return code, additional details will be listed in the RDPRINT output. Make corrections to any source files with errors using the information provided by the SUMMARY DD and RDPRINT DD output. Once you have corrected the source files, queue them for analysis, start the MVS postprocessor, and submit the job again.

Distributed assets

1. Scan Java™ bytecode, not source code, because the product's analysis of Java bytecode is more complete and accurate. Where possible, consider scanning packaged EAR or WAR files instead of scanning the sources that compose those files.
2. Scan one instance of each application. Even when an application is deployed on multiple servers, consider scanning just one copy of it to reduce the time it takes to analyze the distributed artifacts.
3. Do not scan the same physical file or folder from multiple scan roots. Doing so may cause unexpected errors in the distributed postprocessing phase.

4. Run the distributed scanner on a machine with a fast connection to the WebSphere Studio Asset Analyzer database. It is better to scan remote servers from a machine that is close to the database than to scan local assets on a machine with a slow connection to the database.
5. Carefully set the WebSphere Application Server scan root parameters. These parameters (WAS home, Java home, and so on) are used by the Pattern Detector component, which attempts to resolve JDBC, JMS, and JCA calls in Java enterprise applications. The Pattern Detector relies on the accuracy and completeness of these parameters.
6. Pay attention to the GranularityLevel setting in the AssetLocator.cfg file. This setting determines the depth of distributed bytecode asset scans. GranularityLevel=Coarse scans to the level of class file assets, while Granularity=Fine analyzes methods and fields in class files.

Other considerations

- During the Pattern Detector phase of distributed scanning, the scanner may periodically use all the available CPU resources. Consider running the scanner on a dedicated machine or at times when the CPU is available. This phase may take a relatively long time to complete (even several hours), depending on the size of the applications that are being scanned.
- WebSphere Studio Asset Analyzer uses the same algorithms as WebSphere Application Server 6.0 to extract and analyze EAR and WAR files. If the distributed scanner reports errors during the extraction or initial analysis of an EAR or WAR file, try deploying that application on WebSphere Application Server. If the application cannot be successfully deployed on WebSphere Application Server, it is not possible to scan it into WebSphere Studio Asset Analyzer in its packaged form. If you still want to scan it, consider renaming the EAR or WAR file to a .JAR extension, or manually extracting its content to a directory and scanning that instead.

Setting location of Assembler load library

The Assembler scanner uses a combination of the HLASM translator (with options NODECK, NOOBJ, ALIGN, and ADATA) and an ADATA interpreter. By default, the standard STEPLIB, JOBLIB, LINKLIST and LPA search paths are used to locate the HLASM load library. If this default is not appropriate for your installation, set a value for the HLASM library by using Option 0, Set up, of the product ISPF interface.

The Assembler support is meaningful yet less robust than the languages that provide more information in ADATA, such as COBOL and PL/I. Therefore, some options of WebSphere Studio Asset Analyzer are limited.

RELATED ONLINE HELP

Taking an inventory > Taking an inventory of MVS assets > Scanning source code > Scanning Assembler code

Taking an inventory > Taking an inventory of MVS assets > Interpreting inventory results > Assembler items

RELATED REFERENCES

Chapter 3. Defining product defaults, *Configuration and Migration Guide*

Handling of special JCL

The JCL scanner processes each step in the JOB looking for EXEC (EXEC PGM=) and data definition (DD) statements. These statements are processed in the following manner:

1. For each EXEC PGM=*program_name*, a run unit is created and added to the inventory, if it does not already exist. The run unit name is *program_name*.
2. For each DD statement, the positional parameters (such as *, DUMMY) and keyword parameters (such as DCB=, DSN=, OUTPUT=) are analyzed to capture data set attribute information.

The following run unit and DD names are of special interest and require the described additional analysis:

DFSRRC00 (Batch IMS)

Examine the following subparameters for DFSRRC00:

1. Examine the first subparameter for a type indication:
    a. If the type is BMP, look at the fourteenth subparameter for the IMS subsystem name:
        1) If the subparameter is found, add the IMS subsystem to the inventory, if it does not already exist.
        2) If the subparameter is not found, the default IMS subsystem name is NULL, and it is not added to the inventory.
    b. If the type is not BMP (for example, DLI), look at the twelfth subparameter for the IMS subsystem name:
        1) If the subparameter is found, add the IMS subsystem to the inventory, if it does not already exist.
        2) If the subparameter is not found, the default IMS subsystem name defaults is NULL, and it is not added to the inventory.
2. Examine the second subparameter for a run unit name:
    a. If the subparameter is found, the run unit name is the subparameter name. Otherwise, the default name of DFSRRC00 is used.
        1) The run unit is added to the inventory, if it does not already exist.
        2) An invocation reference is added to the run unit, if it already exists.
    b. If the subparameter is DSNMTV01, additional analysis is needed. (When IMS programs need to access DB2, program DSNMTV01 is invoked.) WebSphere Studio Asset Analyzer attempts to read the first record in the DDITV02 data set to determine the following additional information. The DDITV02 DD data set should be available to the JCL scanner, so that the scanner can determine the actual program names. While this is optional, WebSphere Studio Asset Analyzer cannot determine this information if the data set is not available.
        1) Examine the first subparameter for a DB2 system name:
            a) If the subparameter is found, the DB2 system name is added to the inventory, if it does not already exist.
            b) If the subparameter is not found, the default DB2 system name is NULL, and it is not added to the inventory.
        2) Examine the ninth subparameter for a run unit name:

        a) If the subparameter is found, the run unit name is the subparameter name.
        b) If the subparameter is not found, the default run unit name of DSNMTV01 is used.
3. Examine the third subparameter for a PSB name:
    a. If the subparameter is found, attach the PSB to the run unit.
    b. If the subparameter is not found, the default PSB name is NULL, and no PSB is attached to the run unit.

All other parameters are ignored.

IKJEFT01, IKJEFT1A and IKJEFT1B (Batch TSO)

WebSphere Studio Asset Analyzer attempts to retrieve data from the SYSTSIN DD to determine the following additional information. The SYSTSIN DD data set should be available to the JCL scanner, so that the scanner can determine the actual program names. While this is optional, WebSphere Studio Asset Analyzer cannot determine this information if the data set is not available.

1. Locate SYSTEM option of the TSO command DSN for reference to a DB2 system name:
    a. If the command is found, the DB2 system name is added to the inventory, if it does not already exist.
    b. If the command is not found or the name can not be determined, the DB2 system name is DSN, and it is added to the inventory, if it does not already exist.
2. Locate the first occurrence of the DSN subcommand RUN PROGRAM for reference to a run unit name:
    a. If the subcommand is found, the run unit name is added to the inventory, if it does not already exist.
    b. If the subcommand is not found, the default run unit name defaults is NULL, and it is not added to the inventory.
3. Locate the PLAN option of the RUN subcommand for reference to a plan name for the run unit:
    a. If the option is found, attach the plan name to the run unit.
    b. If the option is not found, the default plan name is the run unit name, and it is attached to the run unit.

All other commands and options in the SYSTSIN DD data set are ignored.

Started tasks: The JCL scanner does not currently provide support for JCL procedures that are invoked as started tasks. While any procedure can be started this way, WebSphere Studio Asset Analyzer takes an inventory of procedure files that are used in this fashion as type ″JCL procedure″. They will be included in the count of Unreferenced assets. (From the browser interface, see **Database > Unreferenced assets > JCL files**.) To resolve the unreferenced asset, you should create a dummy JOB that references the PROCs, and submit the file for inventory. The following example is such a job:

```
//DUMPROCR  JOB ,'STARTED TASKS',CLASS=A,
//        MSGLEVEL=(1,0),MSGCLASS=H,NOTIFY=&SYSUID
//*
//PROCLIB JCLLIB ORDER=proclib_name
//*
```

```
//STEP1   EXEC PROC=proc_name_1
//STEP2   EXEC PROC=proc_name_2
//STEP2   EXEC PROC=proc_name_3
//
```

Setting up to scan CICS

The following sections describe the best practices for setting up CICS assets for taking an inventory of CICS region assets.

Extracting CICS asset information

The CICS region scanner uses the CICS utility DFHCSDUP to extract connection, transaction, program, map set, web service, URI map, pipeline, file, and data set resource definition information from the CICS System Definition (CSD) file. This is a static analysis; there is no requirement for the CICS region to be up and running during the scan. (The region can be running.)

By default, all CICS groups in the CSD are considered in scope for analysis and data collection. You should review the CICS groups that are defined for the regions of interest and limit analysis to only those of interest, as described below. For example, you will likely want to exclude those groups that are provided with the CICS IVP samples (that is, DFH*).

You can log on to the CICS region of interest and identify the defined CICS groups. For example, using the CICS CEDA DI(splay) Group will provide information about the groups and their resources. Make note of any groups of particular interest.

The CICS region scanning is performed by a batch job. This JCL is automatically generated when you run the corresponding option from the WebSphere Studio Asset Analyzer ISPF menu. The JCL is presented in an edit session. You can filter by groups with a simple modification. For example, delete the GRP * entry and replace it with individual GRP rows, such as the following example:

```
//*-----------------------------------------------------------------*
//* DMH3000S - CICS REGION ANALYZER
//*-----------------------------------------------------------------*
//DMHCICSP EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//STEPLIB  DD  DSN=DMH.V5R1M0.SDMHLOAD,DISP=SHR
//         DD  DSN=CICSVS.TS230.SDFHLOAD,DISP=SHR
//         DD  DSN=CEE.SCEERUN,DISP=SHR
//CESYSIN  DD *
SITE your_site_name
REGN CICS23
CSD  CICSEC.TS230.CICS.DFHCSD
GRP ABCTEST
GRP DEFPROD
GRP GHIMODL
SJCL SYS1.PROCLIB(CICTS23S)
/*
```

Each group name must start in column six and have a maximum length of eight characters.

RELATED TASK HELPS
Taking inventory > Scanning resources > Scanning a CICS region

RELATED ONLINE HELP

Taking an inventory > Taking an inventory of MVS assets > Scanning CICS regions

RELATED REFERENCES

Chapter 4. The resource definition batch utility DFHCSDUP,
CICS Transaction Server for z/OS® CICS Resource Definition Guide

Chapter 2.8. CEDA--resource definition online,
CICS for MVS/ESA™ CICS-Supplied Transactions

Scanning CICS startup JCL

The CICS region scanner uses the JCL scanner to optionally analyze the CICS startup JCL. The JCL scanner does not currently provide support for JCL procedures that are invoked as started tasks. For CICS startup, if the EXEC PGM=DFSHIP is in a PROC, that is, invoked as a started task, and is not part of a JOB, the JCL scanner generates the following messages because it is expecting the PROC to be in the context of a JOB:

```
DMH1201I Container=proclib_name File=proc_name
DMH1407E Instream PROC PEND not found before end of file
DMH2032I Return code from JCL scanner 8
```

Analysis of the CICS startup JCL and PROCs is optional, primarily for the purpose of identifying additional data sets (such as overrides of those found in the CICS CSD). If your CICS startup is a started task, you should create a dummy JOB that references the PROC, and take an inventory of the file. The information from the invocation of DFHSIP is then available in the inventory. The following example shows such a job:

```
//DUMSTRTJ  JOB ,'CICS START-UP',CLASS=A,
//        MSGLEVEL=(1,0),MSGCLASS=H,NOTIFY=&SYSUID
//*
//PROCLIB JCLLIB ORDER=proclib_name
//*
//STEP1   EXEC PROC=proc_name
```

Arranging access to a CSD

Use the techniques in this section to handle unavailable CSDs, such as the following situations:

- If the production machine with a CSD is unavailable, consider REPROing a copy of the file and relocating it to a machine to which WebSphere Studio Asset Analyzer has access.
- If access to the CSD is strictly prohibited, determine if browsing the resource definition files is permissible. You can use the resource definition files to extract data of interest and manually build a CICS region import file. Import records are output by a variety of WebSphere Studio Asset Analyzer scanners and analyzers and are the metadata input for the database loader.

  Thus, you use the import record definition information to manually build the import records. Once you have created the import records, perform the On-demand load option of the WebSphere Studio Asset Analyzer ISPF menu to process the records and load the database with the CICS region metadata. There is potential for hundreds or thousands of assets, so use this approach only as a last resort.

  The following simple example illustrates using this sample resource definition file:

```
DEFINE FILE(COMPFILE) GROUP(PDPAK)
DESCRIPTION(Company file for Problem Determination tools)
       DSNAME(WSAAUSR.PDPAK.COMPFILE) RLSACCESS(NO) LSRPOOLID(1)
       READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS) STRINGS(1)
       RECORDSIZE(90) KEYLENGTH(20) STATUS(ENABLED) OPENTIME(FIRSTREF)
       DISPOSITION(SHARE) DATABUFFERS(2) INDEXBUFFERS(1) TABLE(NO)
       MAXNUMRECS(NOLIMIT) UPDATEMODEL(LOCKING) LOAD(NO)
       RECORDFORMAT(V) ADD(YES) BROWSE(YES) DELETE(NO) READ(YES)
       UPDATE(YES) JOURNAL(NO) JNLREAD(NONE) JNLSYNCREAD(NO)
       JNLUPDATE(NO) JNLADD(NONE) JNLSYNCWRITE(YES) RECOVERY(NONE)
       FWDRECOVLOG(NO) BACKUPTYPE(STATIC)
DEFINE FILE(CUSTFILE) GROUP(PDPAK)
DESCRIPTION(Customer file for the Problem Determination tools)
       DSNAME(WSAAUSR.PDPAK.CUSTFILE) RLSACCESS(NO) LSRPOOLID(1)
       READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS) STRINGS(1)
       RECORDSIZE(136) KEYLENGTH(81) STATUS(ENABLED) OPENTIME(FIRSTREF)
       DISPOSITION(SHARE) DATABUFFERS(2) INDEXBUFFERS(1) TABLE(NO)
       MAXNUMRECS(NOLIMIT) UPDATEMODEL(LOCKING) LOAD(NO)
       RECORDFORMAT(V) ADD(YES) BROWSE(NO) DELETE(NO) READ(YES)
       UPDATE(YES) JOURNAL(NO) JNLREAD(NONE) JNLSYNCREAD(NO)
       JNLUPDATE(NO) JNLADD(NONE) JNLSYNCWRITE(YES) RECOVERY(NONE)
       FWDRECOVLOG(NO) BACKUPTYPE(STATIC)
DEFINE TRANSACTION(MYTD) GROUP(PDPAK)
DESCRIPTION(Start Trader application VSAM version)
       PROGRAM(MYTRADMV) TWASIZE(0) PROFILE(DFHCICST) STATUS(ENABLED)
       TASKDATALOC(BELOW) TASKDATAKEY(USER) STORAGECLEAR(NO)
       RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
       ROUTABLE(NO) PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO)
       RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
       CONFDATA(NO) ACTION(BACKOUT) WAIT(YES) WAITTIME(0,0,0)
       RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(TDB2) GROUP(PDPAK)
DESCRIPTION(Start Trader application DB2 version)
       PROGRAM(MYTRADMD) TWASIZE(0) PROFILE(DFHCICST) STATUS(ENABLED)
       TASKDATALOC(BELOW) TASKDATAKEY(USER) STORAGECLEAR(NO)
       RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
       ROUTABLE(NO) PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO)
       RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
       CONFDATA(NO) ACTION(BACKOUT) WAIT(YES) WAITTIME(0,0,0)
       RESSEC(NO) CMDSEC(NO)
```

You could create an import file that contains the following records to represent the sample CICS resources:

```
FMT REGION 1
TOOL 0001000011
SITE your_site_name
START REGION your_region_name
RU MYTRADMV
TRAN MYTD PDPAK
RU MYTRADMD
TRAN TDB2 PDPAK
DD START COMPFILE
DD GROUP PDPAK
DD SEQ 1
DD DSN WSAAUSR.PDPAK.COMPFILE
DD END
DD START CUSTFILE
DD GROUP PDPAK
DD SEQ 1
DD DSN WSAAUSR.PDPAK.CUSTFILE
DD END
END REGION
```

Related online help

Administrative information > Import File Description

Scanning IMS subsystem assets

Taking an inventory from an IMS subsystem involves a series of scanners and analyzers to capture information about data set, transaction, and control block assets (DBDs: Database Descriptions; PSBs: Program Specification Blocks; and ACBs: Application Control Blocks).

All but one IMS option perform static analysis. There is no requirement for the IMS control region to be up and running at the time of analysis, except if you are running Option 2.3.A, **Scan IMS commands issued in batch**. (The region can be running.)

All IMS subsystem scanning is performed by batch jobs. The JCL is automatically generated when you run the corresponding option from the WebSphere Studio Asset Analyzer ISPF menu. You should carefully review the options that are presented for each IMS subsystem asset type. The asset type suboptions are mutually exclusive; you should not run each and every option that is displayed. For example run only one of each of the following options:

- DD and database data set information from RECON data sets or the Dynamic Allocation load library
- Transaction and run unit information from Stage 1 or Stage 2 or MODBLKS DFSSMB0x data sets or commands issued in batch
- DBD and PSB control block information from DBDLIB and PSBLIB libraries or ACBLIB library

Depending on your IMS installation and configuration, not all suboptions on the ISPF interface may be available at your site. For example, the IMS Library Integrity Utilities (LIU) are used to reverse engineer IMS binaries. If LIU is not available at your site, you can collect inventory for IMS DBDGEN and PSBGEN source files by using the browser interface. However, the PSB-to-run unit matching will work only if the IMS Binaries (either ACBLIB or PSBLIB, but not both) are scanned. If the PSB source is scanned, PSB-to-run unit matching will not occur. Use the following strategy to determine how best to scan IMS control blocks:

MAP 0010: Deciding how to scan IMS control blocks

001

Is the ACBLIB available?
Yes No

002

Is the PSBLIB available?
Yes No

003

– Scan the PSB source and proceed to Step 005.

004

– Scan the PSBLIB and proceed to Step 005.

005

Is the DBDLIB available?
Yes No
Scan the DBD source.

006

– Scan the DBDLIB

007

– Scan only the ACBLIB.

RELATED TASK HELPS
Taking inventory > Scanning resources > Scanning an IMS subsystem

RELATED ONLINE HELP
Taking an inventory > Taking an inventory of MVS assets > Scanning IMS subsystems
Taking an inventory > Taking an inventory of MVS assets > Scanning IMS subsystems > Analyzing IMS control blocks.

RELATED REFERENCES
Chapter 6. DBD/PSB/ACB reversal,
IMS Library Integrity Utilities for z/OS User's Guide

Scanning DB2 Catalog Assets

Taking an inventory of a DB2 catalog is done by a series of batch jobs that capture information about tables, views, columns, and stored procedures for a given DB2 system. This information is read from the SYSIBM.SYS* catalog tables.

The JCL is provided in the *yourHLQ*.CONFIG.CNTL data set and includes the following activities:

- Report table and stored procedure schema names that are found in the catalog.
- Scan the catalog for schemas of interest and gather metadata.
- Load metadata into the database.
- Relate source SQL references to DB2 catalog metadata.

The amount of catalog data that is collected is controlled through the use of a scan filtering mechanism in the DMH565I parameter file in the *yourHLQ*.CONFIG.DATA data set. By default, the scan filter is set to '*', which indicates that data for all schemas found in the catalog is collected. This can be time consuming and involve a substantial amount of space, depending on the number of schemas and volume of data in the DB2 catalog.

You should carefully review the information that is provided in the report step, identifying schemas that are of interest to WebSphere Studio Asset Analyzer users and modifying the scan filter accordingly. You can use an inventory approach that is based on incremental scans and loads using different scan filters over time to build metadata for the catalog assets.

RELATED TASK HELPS
Taking inventory > Scanning resources > Scanning a DB2 system

RELATED REFERENCES
Appendix D. DB2 catalog tables, *DB2 Universal Database™ for OS/390® and z/OS (V7)*
Appendix F. DB2 catalog tables, *DB2 Universal Database for z/OS (V8) SQL Reference*
Appendix. Additional information for DB2 SQL > DB2 catalog tables, *DB2 Version 9.1 for z/OS SQL Reference*

Chapter 4. Checking inventory status

You should monitor analysis results during the Inventory activities and address issues as you go along, as opposed to leaving them all for review and resolution at a later time. You can easily verify the status of the metadata and resolve issues by verifying each container and application after the metadata is loaded into the database.

Checking status on summary and details pages

You can view the **Analysis status** on the asset-specific summary and details pages of files, batch jobs, programs, and run units to check the analysis status of particular batch jobs, programs, and run units. Use the **Analysis status** field of **Advanced search** on those pages to filter assets that have the following analysis status values:

Completed
Queued requests completed without errors

Error Queued requests completed with errors

Incomplete
Queued requests are outstanding and are being processed

″Incomplete″ is displayed if you are exploring the inventory while the queue processor is running. If you are an administrator, you can confirm this by clicking **Database > Analysis queue** and confirming the **MVS postprocessing status** field contains the following message:

```
!  DMH5349W - MVS postprocessor is running.  Search results may not be accurate
```

An advantage to using the summary and details pages to check status, rather than the Database pages as described in the next section, is that users without administrative authority can identify asset problem areas to assist with resolving issues. The following example gives such a scenario:

1. From a Batch job, Program, or Run unit summary page, the user can select the following **Advanced search** criteria, and then submit the search:
   - An application from the **Application** list
   - **Error** from the **Analysis status** list.
   - Possibly additional filtering by **Language** and **Program type** on the Program summary page

   The refreshed summary page displays those assets that are in error state.
2. Click the **Analysis status** of an asset to open the details page.
3. Review the Analysis Messages.

While non-administrators cannot perform additional tasks to actually resolve the problem (such as taking an inventory for missing assets or assigning or updating concatenation sets), they can assist with identifying the cause of the reported errors.

No asset with a language other than UNK? should be in incomplete state following normal completion of the analysis queue processor. In the unlikely event that ″Not running″ is indicated on the Analysis queue information page, you should investigate why the asset is in Incomplete state. Review the inventory JOB output, especially the JES messages and RDPRINT output from WebSphere Studio Asset Analyzer for indications of a problem, such as an abend or other system error. If the JOB is no longer available, requeue the file for analysis and monitor the results.

Checking status on the Database pages

While you can determine the analysis status of assets from the summary and details pages, of special interest are the following pages that are available from the **Database** menu:

Errors The Errors page provides information about errors for MVS assets that were detected during inventory and postprocessing. The header section of the Errors page reports error rates for batch jobs, programs, and run units. These percentages represent the ratio of files (batch jobs and programs) and run units that are not in complete state to the total number of files or run units in the inventory. The goal is to get the error rates as close to zero as possible.

Click a percentage in the header section of the Errors page to open a summary errors page for the particular asset type (that is, batch job, program or run unit). The Analysis status breakdown section shows counts by status. Click the count for Error to open an asset summary page that contains results for the particular asset type. From there you can click the asset or **Error** in the **Analysis status** column to open the details page for that asset and investigate the **Analysis Messages**. Refer to the WebSphere Studio Asset Analyzer *Messages and Codes Guide* to understand the error. Use the User response section of the message to resolve the situation.

The header section on the Errors page also reports **Activity log errors**. Click the number of errors to open the Activity log error summary page. This page presents the broadest view of all the errors that were discovered during inventory and postprocessing. Use the Advanced search criteria and column sorting functions to manage the data.

Below the header section on the Errors page is a series of tables that show errors by category. The most common errors are related to missing inventory components such as entry points (program source) or included files (JCL PROCs and program copy files). If you address these errors first, you will build a more complete inventory and will avoid potential downstream issues, such as incomplete search and impact analysis results.

Unreferenced assets

The Unreferenced assets page provides information about MVS assets that have been scanned into the inventory but are not referenced. The header section of this page reports counts for the following unreferenced components:

JCL files

Unused PROCs and INCLUDE files

Online data stores

Unused Data stores from the CICS region scanner and IMS database data set scanners

Programs

Programs for which there are no invoking batch jobs or transactions

Run units

Run units not referenced by any batch job or transaction

Source include files
Unused COBOL COPYs, PL/I INCLUDEs, Assembler macros

Click a nonzero count to open a summary page for that asset. You can further filter the results by using additional **Advanced search** criteria on the summary page, such as **Application** for programs. Keep **Not referenced** selected on the Online data store, Program, and Run unit summary pages to stay in the context of unreferenced assets. (The File summary pages that are presented for JCL files and Source include files use a default set of criteria to produce the list of unreferenced results. These criteria exclude reporting any JCL files that have batch jobs or job steps matches or source include files that have program or include matches. The criteria are shown in the displayed search criteria.)

Unreferenced assets are most often the result of an incomplete inventory. Review each entry and determine if it is obsolete and can be deleted from the inventory, or if additional inventory needs to be taken. You need to consider the following two additional unreferenced components:

Unreferenced JCL Procedures
The JCL scanner does not currently provide support for JCL procedures invoked as started tasks. PROCs of this nature are included in the count of unreferenced JCL files. To report only unreferenced JCL procedures, set the **File type** attribute for **Advanced search** on the File summary page to "JCL procedure". Review each entry and determine if any are JCL procedures invoked as started tasks. If so, to have these procedures not be flagged as unused, you should create a dummy JOB that references the procedures, and then submit the file for inventory.

Unreferenced run units
The following situations add to the count of unreferenced run units:

- Run units are created for all primary entry points to resolve external calls and to enable finding connection points for e-business applications. As a result, some run units might not have an invoking batch job or transaction. These run units are reflected in the count of unreferenced run units on the Unreferenced assets page.
- When a run unit is generated for the main entry point, and that entry point is later aliased to some other entry point, the original run unit remains in the inventory and is reflected in the count of unreferenced run units. For example, the following code results in a run unit named SAMPABC:

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  SAMPABC.
...
```

  It is later discovered that the link-edit step for SAMPABC renames the load module to TESTABC. Therefore, you should add TESTABC to the *yourHLQ*.CONFIG.DATA(ALIASES) file as an alias for SAMPABC. With subsequent processing, a run unit is created for TESTABC. Run unit SAMPABC is extraneous and can be deleted from the inventory.

  You can avoid this situation by identifying alias entry points before you take an inventory. Identify the aliases by manually adding them to the ALIASES file.

- A run unit is generated for the first CSECT that is found. Later, the assignment of a concatenation set locates a missing macro that contains another CSECT, which displaces the original CSECT as an alternate CSECT. But the run unit for the original CSECT remains in the inventory and is reflected in the count of unreferenced run units.

  For example, the following code results in a run unit named DFSISISO:

  ```
  ISIS0    TITLE 'CTL, ISI USER SECURITY'
           XYZMACRO ,PUT AS FIRST CSECT
           EJECT
           REQUATE
  DFSISIS0 CSECT
  ```

  It is later discovered that XYZMACRO is reported as not found. A concatenation set is assigned and analysis identifies a CSECT in the XYZMACRO named XYZIVP1, resulting in a run unit named XYZIVP1. Run unit DFSISISO is extraneous and can be deleted from the inventory. You can avoid this situation by assigning concatenation sets when you take an inventory.

Related task helps

Taking inventory > Refining your inventory >
Finding COPY and INCLUDE files

Taking inventory > Refining your inventory >
Defining alias names for programs

Related online help

Help for product pages > Database > Errors

Help for product pages > Database > Errors > Activity log error summary

Help for product pages > Database > Unreferenced assets

Chapter 5. Handling concatenation sets

A concatenation set is an ordered list of containers to be searched when resolving references to included source during analysis scans. The types of included source searched for are Assembler macros, COBOL copybooks, JCL PROCs, PL/I includes and SQL INCLUDEs. To best represent your processing environment and inventory, you should locate existing compilation jobs to determine the actual libraries and search order that are appropriate for the source of interest. In general, your installation's existing JOBLIB, STEPLIB, and SYSLIB concatenations will yield valuable information that you can use for building the appropriate WebSphere Studio Asset Analyzer concatenation sets.

Constructing concatenation sets

You can construct concatenation sets by using the browser interface wizard or having them be generated automatically. You can assign a concatenation set at the following four ordered levels of the asset hierarchy to simplify include file resolution:

1. Site
2. Container
3. Application
4. File

If a concatenation set is required to resolve included files, the higher in the hierarchy above that you assign it, the better. That is, assigning a concatenation set to a site encompasses the most file coverage, a specific file the least. Assigning a concatenation set at the file level is the most efficient from a pure processing standpoint, but not from a practical use perspective. You would typically assign a concatenation set to a particular file only to override the default concatenation set search order. Assigning a concatenation set at the container level is by far the most common level for WebSphere Studio Asset Analyzer users.

You need administrative authority to create, update, or assign concatenation sets with the wizard. You can manage the sets in the following way:

- Create or update a concatenation set from the concatenation set details page.
- Create, update, and assign at the container level from the Assign default concatenation sets page of the Inventory wizard.
- Assign to a site, application, container, or file from the appropriate details page by using **Assign default concatenation sets** from the **Actions** list. If any concatenation sets are assigned, they are shown in the **Concatenation set assigned** or **Default concatenation sets** field of the header section on the asset-specific details pages.

Adhere to the following rules for concatenation set names:

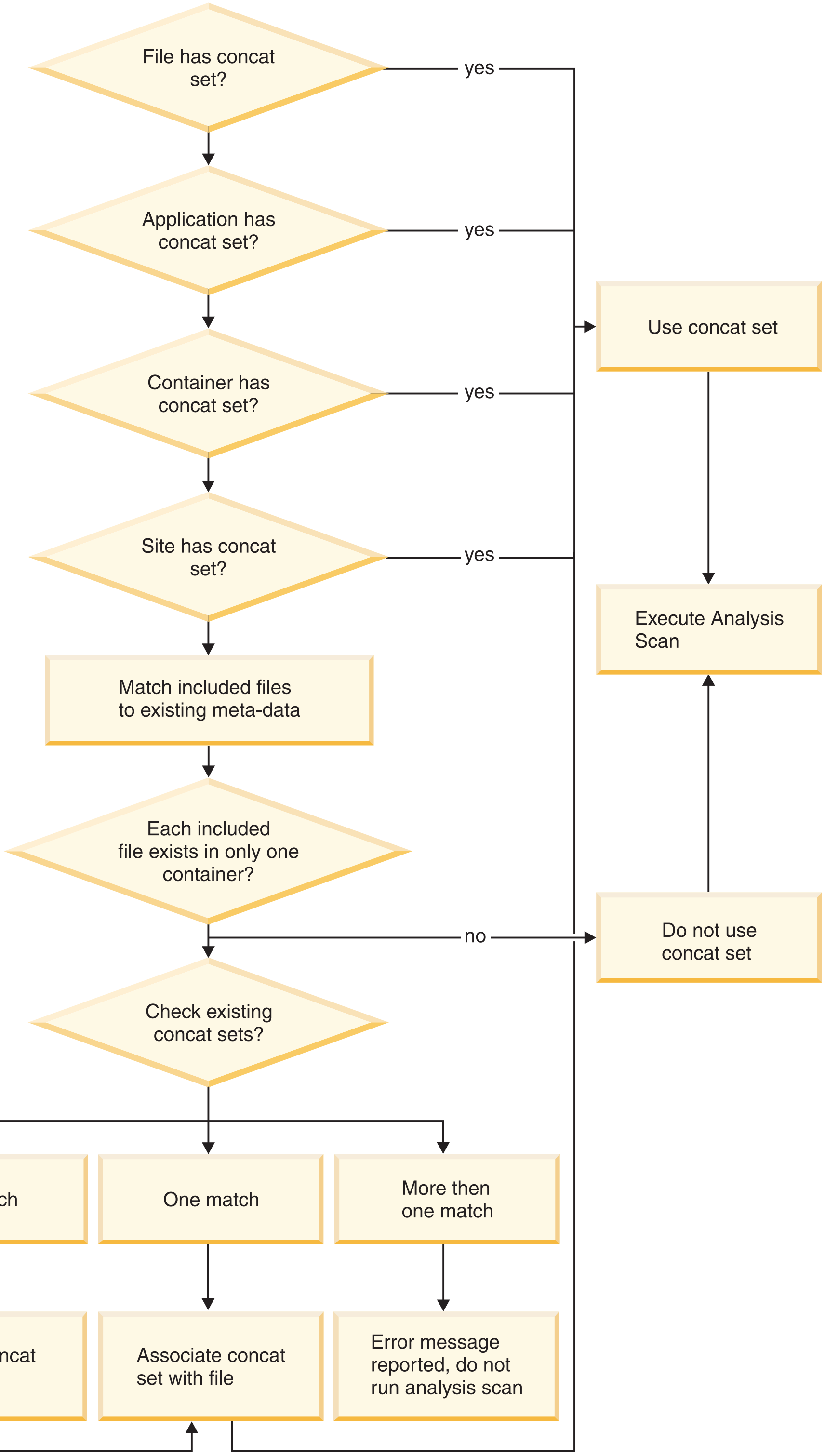
- Uppercase character string of not more than eight characters
- Alphabetic characters, #, @, $, and numeric digits
- The first character cannot be numeric

You should define meaningful naming conventions for the concatenation sets that you create. For example, use application or ownership acronyms, test or build designations, department abbreviations, and so on. By default, concatenation sets that are generated automatically by WebSphere Studio Asset Analyzer are prefixed by "DMH" followed by a sequential number (that is, DMH1, DMH2, ...).

Concatenation sets are automatically generated when you do not specify a concatenation set for an analysis scan. In this case, WebSphere Studio Asset Analyzer attempts to find, or create, an appropriate concatenation set based on the containers and files that are already represented in the inventory. If successful, the found or created concatenation set is assigned at the file level, so that it can be used immediately on any subsequent rescans. If analysis is not successful in determining an appropriate concatenation set, the analysis scan can be run without a concatenation set; any missing include files are reported as such.

All concatenation set definitions are stored in the database. Additionally, both user-defined and generated concatenation sets for COBOL, PL/I, and JCL are stored as members in a partitioned data set (PDS) named *&uhlq*.INCL2, where *&uhlq* is the high-level qualifier for user data sets that was specified during product configuration. However, concatenation sets for Assembler are not stored in this PDS. This PDS is maintained by WebSphere Studio Asset Analyzer. You should not modify it, because synchronization with the database definitions could be compromised.

The following diagram is an overview of the rules that are applied by WebSphere Studio Asset Analyzer to resolve concatenation sets:


File has concat set?
yes
Application has concat set?
yes
Container has concat set?
yes
Site has concat set?
yes
Use concat set
Execute Analysis Scan
Match included files to existing meta-data
Each included file exists in only one container?
no
Do not use concat set
Check existing concat sets?
No match
One match
More then one match
Create concat set
Associate concat set with file
Error message reported, do not run analysis scan

RELATED TASK HELPS

Taking inventory > Refining your inventory > Finding COPY and INCLUDE files

RELATED ONLINE HELP

Taking an inventory > Taking an inventory of MVS assets >

Scanning source code > Creating a search order for included source
Help for product pages > Database > Inventory >
Assign default concatenation sets

An example

This example illustrates an assignment of concatenation sets for the following containers:

- Container A - some files in this container reference COPY1, COPY2, COPY3.
- Container B - some files in this container reference COPY1.
- Container C - some files in this container reference COPY4, COPY5, COPY6.

Given these containers, you can create one or more concatenation sets, depending on the factors that are required to locate all the include files of interest, such as your standards, the number of containers, the number of files per container, and multiple versions of files. For example, you might create the following concatenation sets:

CSET1
List the containers where COPY1, COPY2, COPY3, COPY4, COPY5, and COPY6 are located. Assign CSET1 as the default concatenation set for containers A, B, and C.

CSET2
List the containers where COPY1, COPY2, and COPY3 are located. Assign CSET2 as the default concatenation set for container A or both containers A and B.

CSET3
List the container where COPY1 is located. Assign CSET3 as the default concatenation set for container B.

CSET4
List the containers where COPY4, COPY5, and COPY6 are located. Assign CSET4 as the default concatenation set for container C.

If the location of included source is compartmentalized into a small number of containers, then CSET1 (that is, a single concatenation set) can readily apply as the default for the entire group of source containers. However, as an example, if COPY1 has different versions, then a minimum of two concatenation sets would be required, such as CSET1 (for COPY1, COPY2, COPY3, COPY4, COPY5, and COPY6) and CSET2 (for the second version of COPY1).

Avoid automatic generation

You should establish concatenation sets early and explicitly assign them to assets. This avoids having WebSphere Studio Asset Analyzer expend additional resources searching the database for existing containers and concatenation sets that best match the requirements for resolving the program source that is being scanned. Assigning concatenation sets also avoids activity that automatically occurs in the absence of an assigned concatenation set.

Complications can occur if like-named include files reside in multiple containers. It is important to understand what versions of particular files apply to what containers. Determining this information prior to the inventory of the program source code resolves the dilemma that is created when WebSphere Studio Asset Analyzer tries to determine the appropriate concatenation set. When more than one file with the same type, language, and name is encountered, the analyzer does not guess. Rather, the analysis status for the including program is reported as "Error", and the included files are reported as missing, with resolution left to you.

RELATED ONLINE HELP

Help for product pages > Database > Inventory > Assign default concatenation sets

Handling Assembler and JCL

Assembler: In the absence of an assigned concatenation set, the Assembler scanner uses SYS1.MACLIB to locate any referenced MACROs and INCLUDEs.

JCL: The JCL scanner processes JCLLIB ORDER=(*library1* [[,*library2*] ...]) statements in the same manner as JES. The libraries that are found in the JCLLIB statement are the default search list and are prepended to the assigned concatenation set. In the absence of an assigned concatenation set, these libraries are used to resolve INCLUDE and PROC references. If access to the libraries that are indicated in JCLLIB statements are RACF-protected or otherwise restricted for users who are taking an inventory, verify that you have configured the JCL analyzer to ignore the libraries that are referenced in the JCLLIB statement, and to continue processing.

RELATED REFERENCES

Chapter 3. Defining product defaults, *Configuration and Migration Guide*

Chapter 6. Managing applications

Applications are user-defined collections of assets. These collections provide a mechanism to easily and quickly understand a subset of logically related assets. Assets can be associated with 0 to *n* applications.

Applications are created in one of the following ways:

- The **Create a new application** action from the Application summary page
- **Assign to a new application** option during inventory of MVS assets
- The one-to-one relationship that is established for distributed scan root to application (that is, default application for crawl root)

You should take advantage of applications in WebSphere Studio Asset Analyzer for the following compelling reasons:

- You can select the application name as an advanced search criteria on the summary pages for all distributed assets (except unresolved assets) and MVS assets (except literals). This provides a powerful filtering mechanism for targeting search results.
- You can use applications to contain the breadth of an impact analysis.
- You can use applications to partition aggregate statistics for the assets that comprise an application.
- You can use parent applications to aggregate the assets of all descendent applications.

RELATED TASK HELPS
Using applications > Understanding applications

Creating and assigning applications

You should carefully consider the following aspects of your applications:

- Application-naming conventions
- Planning the parent-child-grandchild hierarchies early in the process
- Always specifying the appropriate application during inventory. It is much easier to assign an application during inventory than after inventory. CICS and IMS transactions are the exceptions, because there is no wizard for assigning them to applications. You must assign transaction assets to their respective applications after the database load.

Adhere to the following rules for your application names:

Application name
Limited to sixteen characters with no intervening white space, but the underscore character (_) is permitted.

Application full name
Free-form text less than 255 characters in length

Application description
Free-form text less than 257 characters in length

You can assign MVS assets to an application in the following ways:

- Automatically, by using the Assign to application page of the Inventory wizard
- Manually from the Application details page. Select files and transactions after they have been scanned into the database, and then use the **Assign assets** action.
- ChangeMan only: A corresponding application with all the ChangeMan parts is automatically created and updated.

After you have taken an inventory of MVS containers, if you chose to create an application in the Inventory wizard, by default, the Application summary page displays ″Created by the Inventory page″ in the **Description** column for the application. You should update the **Name** and **Description** columns with information that is meaningful for WebSphere Studio Asset Analyzer users. Use the **Update application definition** action on the Application details page.

If you are assigning source file assets after taking an inventory, you should use the **Container** field of **Advanced search** on the Application details: Assign File assets page to simplify the selection process.

You can also remove assets from an application by using the **Remove assets** action on the Application details page. This is different from, and should not to be confused with, deleting assets from the inventory.

You must define child and grandchildren applications as such at the time you create them, by selecting the parent application from the list on the Application definition: Create page of the wizard. There is no mechanism for converting an existing application to a child of another application after it has been established.

RELATED TASK HELPS
Using applications > Adding assets to an application
Using applications > Removing assets from an application

RELATED ONLINE HELP
Help for product pages > Home > Common detail pages > Application
**Help for product pages > Home > Common actions >
Create a new application**
Help for product pages > Database > Inventory > Assign to application

Applications with source management systems

WebSphere Studio Asset Analyzer automatically builds application hierarchies for Software Configuration Management (SCM) and Software Configuration and Library Manager (SCLM) inputs. When you take an inventory, you have the option to select the hierarchy that WebSphere Studio Asset Analyzer creates, create a new application, use an existing application, or to not assign to an application. ChangeMan is the only exception to this selection.

You can consider this subset of assets a snapshot of the application at a point in time; it can change as you rescan, add, or delete assets. You should consider the following aspects of the snapshot:

- Which snapshot you want: development, unit test, quality assurance, or production.
- Integration of code into your SCM or other hooks in your development processes can trigger rescans.

CA-Endevor, CA-Librarian, and CA-Panvalet
Application is user-defined: create or select the application from the list on the Specify the application page of the Inventory wizard when you are taking an inventory.

SCLM SCLM Project identifier is the parent Application.
SCLM Group identifier is a child Application.

For example, given the following SCLM hierarchy:

```
Project = TSTSCLM
Group = RELEASE
```

When you take an inventory, you are prompted to select an application on the Specify the application page of the wizard, where the default selection for this example would be set to the following assignment:

```
<Assign to new "RELEASE" application in "TSTSCLM">
```

Serena ChangeMan
ChangeMan subsystem ID is the parent Application.
ChangeMan application ID is a child Application.

For example, given the following ChangeMan input during inventory:

```
Subsystem Id = SERA
Application Id = DMH1 - WSAA TEST APPLICATION 1
```

You would be presented with the following static fields on the Specify the application page of the wizard:

```
SERA in the ChangeMan Subsystem Id field
DMH1 - WSAA TEST APPLICATION 1 in the ChangeMan Application Id field
```

The options that are available, active SERNET server instances and defined applications, are based on your ChangeMan configuration.

RELATED TASK HELPS

Taking inventory > Scanning resources > Scanning ChangeMan applications
Taking inventory > Scanning resources > Scanning source located on other software configuration management systems

RELATED ONLINE HELP

Taking an inventory > Taking an inventory of MVS assets > Scanning source code > Scanning without the Web browser

RELATED REFERENCES

Chapter 3. Configuring source code management systems, *Configuration and Migration Guide*
Chapter 3. Configuring a user preprocessor, *Configuration and Migration Guide*

Chapter 7. Resolving run units

A run unit is the logical collection of all programs or routines that are capable of being reached through a call or transfer of control when a batch job or transaction runs. WebSphere Studio Asset Analyzer creates a run unit when it does not already exist in the inventory, for the following situations:

- A batch JCL EXEC PGM= statement is analyzed.
- A CICS or IMS transaction is analyzed.
- A primary (first found) entry statement is analyzed in a program.

Resolve types

Each level in the calling hierarchy of a run unit corresponds to one of the following levels:

0 Program with no main program found

1 Main program source

>1 Subprogram source

Each entry point in the calling hierarchy of a run unit has a resolution type that corresponds to one of the following types. The type indicates how the source code was or was not reached.

Table 3. Resolve types

Type	Displayed Resolve type	Analysis status for run unit
1	Resolved by Utility	Completed
2	Resolved by Entry Point	Completed
3	Resolved by Alias	Completed
4	Unresolved - No Entry Point	Error[1]
5	Unresolved - Multiple Entry Points	Error[1]
6[2]	Unresolved - Component Delete By User	Error[1]
7[3]	Resolved by Recursion	Completed

Notes:

1. Is displayed in at least one row of the calling hierarchy when the analysis status is "Error" for a run unit. That is, for a run unit with a multilevel hierarchy, one of these types is displayed for at least one entry point in the calling hierarchy. Other entry points in the hierarchy might have a resolve type of 1, 2, or 3.
2. Occurs when a previously resolved run unit is updated as the result of a program being deleted from the inventory.
3. Is displayed in a row when a call to the same named entry point has been identified earlier in the calling path being traced.

Options and limitations for resolving a missing entry point

You can use the following options to resolve a run unit with a missing entry point:

- **Option:** Locate the missing program source, and submit it for inventory.

- **Option:** Include the program name in the *yourHLQ*.CONFIG.DATA(UTILS) file to resolve the entry point as a utility, and queue it for analysis. This will forgo any further search for program source.

  Limitations: This option will quickly resolve a run unit with missing entry point error messages, however, you must ensure the program is a utility. If the program is not a utility, do not resolve the missing entry point errors by including the program name in your utility file.

  If you use this option, you cannot reverse it simply by removing the entry from the utility file. To reverse this option, you must remove the entry from the utility file and force the rebuilding of the run unit by changing the run unit's analysis status to Incomplete, for example, in order to have the "Resolved by Utility" status removed from the run unit's calling hierarchy.
- **Option:** Include the program name in the *yourHLQ*.CONFIG.DATA(ALIASES) file to resolve the entry point as an alias of another program, if applicable, and queue it for analysis.

  Limitations: This option will quickly resolve a run unit with missing entry point error messages, however, you must ensure the program is an alias. If the program is not an alias, do not resolve the missing entry point errors by including the program name in your aliases file.

  If you use this option, you cannot reverse it simply by removing the entry from the aliases file. To reverse this option, you must remove the entry from the aliases file and force the rebuilding of the run unit by changing the run unit's analysis status to Incomplete, for example, in order to have the "Resolved by Alias" status removed from the run unit's calling hierarchy.

  If renaming is pervasive at your installation and you are familiar with link-edit decks, you can write an external script to generate the information to be loaded in the aliases file. WebSphere Studio Asset Analyzer does not process link-edit decks directly, but it provides the aliases file for mapping between the new and default program names.
- **Option:** Create a dummy stub program to resolve the primary entry point reference, and submit it for inventory. The stub program must have the name of the missing entry point.

  Limitations: By adding the dummy program to the inventory, the missing entry point error will be cleared and no trace will be left of it unless you document the fact that there is no real source for the program. Consider creating a mechanism to remember the list of programs for which a dummy stub was created. For example, write and inventory an additional special-purpose dummy program and name it something like "NOSOURCE." Ensure that all dummy stub programs created to resolve missing entry point errors have a call statement to NOSOURCE. With this mechanism in place, a lookup of the program details for NOSOURCE will list all the calling programs. The result is a list of the programs for which the real source is missing. At the same time, a run unit involving any of the programs for which there is no real source will include the call to the NOSOURCE program.

A run unit error caused by a missing entry point is a detail provided by WebSphere Studio Asset Analyzer to inform you that a program is missing. This is a valuable piece of information about your inventory. Leaving the error is a valid option and can be the best choice. The error serves as a reminder that parts are missing and follow-up action for a permanent resolution might be required.

In some cases, a run unit with a missing entry point might be unreferenced as a result of a deleted asset. For example, it is possible that the asset leading to the generation of the run unit (for example, a batch job, transaction, or program) was mistakenly inventoried and subsequently deleted. In this case, the run unit will appear as removable for users with administrative authority. Carefully determine if it is appropriate to remove the run unit from the remaining inventory.

Resolution during postprocessing

WebSphere Studio Asset Analyzer resolves run units by building call hierarchies for each in the following ways:

1. The run unit name is matched against a list of known utility programs to see if call resolution should be attempted. If the name matches an entry in the utility file, no search for source code is done. (You can add your own utility programs to the list.)
2. The run unit name is matched against a list of known alias names to see if call resolution should be attempted for a name other than the current run unit name. If the name matches an entry in the alias file, the alias name is used as the name of the run unit. (You can add your own alias names to the list.)
3. If no match was found in the utility or alias files, the run unit name is matched against all the entry point names in the inventory. If one or more names match, the search stops. If more than one program name matches, a warning message is generated. (There is no mechanism to support multiple entry points of the same name.)
4. If at least one match is found in steps 1, 2, or 3, then level 1 of the calling hierarchy is created. If no matches are found, level 0 is indicated for the entry point.
5. To complete the call hierarchy entries below the main program, each control transfer (CALL, CICS LINK, CICS XCTL) in a program is processed to resolve the target program of the transfer. If a recursive call is encountered, an informational message is issued for the run unit, and processing continues with the next transfer target.

RELATED TASK HELPS

Exploring assets - MVS > Run unit >
Listing the entry points in a run unit

RELATED ONLINE HELP

Taking an inventory > Taking an inventory of MVS assets >
Interpreting inventory results > Ambiguous run units
Help for product pages > Explore MVS assets >
Summary pages > Run unit

Identifying utility programs

WebSphere Studio Asset Analyzer uses two files for identifying utility and utility-like programs when resolving run unit names. Without regard to Site, these two files are used during postprocessing to decide which called programs should not be candidates for source location when building call hierarchies. The files of interest are found in the following data sets:

- *yourHLQ*.CONFIG.DATA(UTILS)
- *yourHLQ*.SDMHDATA(DMHUTIL)

Both of these 80-byte text files provide a mechanism by which you can easily add and maintain lists of your own and third-party utility programs and subroutines.

To identify utility programs that should resolve run unit names for your installation, you need to update the user-customized file that is used by WebSphere Studio Asset Analyzer during processing, as described in step 1 of "Resolution during postprocessing" on page 39. This list of programs is found in the UTILS file. By default, the UTILS file is empty. New program names can be added to this file starting in column one, with a single entry per line. During subsequent installation of future releases and PTFs for WebSphere Studio Asset Analyzer, the product will not update or overwrite the *yourHLQ*.CONFIG.DATA(UTILS) file.

A similar list of utility programs, used in combination with the user-customized UTILS file, can be found in the DMHUTIL file. This file contains the names of common utility programs and subroutines, such as IEBGENER, IDCAMS, and CBLTDLI. It is shipped with a default set of entries from IBM and others. During subsequent installation of future releases and PTFs for WebSphere Studio Asset Analyzer, the product may update the *yourHLQ*.SDMHDATA(DMHUTIL) file. Therefore, to preserve your updates, apply any modifications to the UTILS file.

The list of program names from the UTILS and DMHUTIL files are concatenated at runtime, with the user-customized list from the UTILS file being in the first position.

As an example, when a called entry point matches a name in the UTILS or DMHUTIL lists, WebSphere Studio Asset Analyzer does not search for source code. That is, the source for the utility is not expected to be found in the inventory. The Run unit details page indicates that the program is a utility, with the **Resolve type** column for the entry point displaying Resolved by Utility, and no hyperlinks are presented for the **Program** and **Source location** columns.

Additionally, you can use the UTILS file for the following programs:

- The source code for the program is not available and likely will never be available for inventory collection.
- There is no interest in tracing beyond the call into the target program. That is, it is satisfactory to register the call but you do not want users to see metadata beyond that point.
- The called program is written in a language that is not supported by WebSphere Studio Asset Analyzer.

You can use the asterisk (*) in the UTILS file as a wildcard to avoid typing large groups of individual names. For example, the DMHUTIL file is shipped with entries for CEE* (Language Environment®) and DFH* (CICS).

You should review and update the UTILS file to include the common utilities and subroutines that your users are most likely to encounter. However, you can update UTILS (and DMHUTIL) at any time; there is no compile or rebind required for WebSphere Studio Asset Analyzer. The next time postprocessing runs, the modified UTILS and DMHUTIL will be in effect.

RELATED TASK HELPS

Taking inventory > Refining your inventory > Defining utility programs

RELATED ONLINE HELP

Help for product pages > Explore MVS assets > Details pages > Run unit

Identifying aliases

WebSphere Studio Asset Analyzer uses two files for identifying aliased entry points. Without regard to Site, these two files are used during postprocessing to decide which called programs should not be candidates for source location when building call hierarchies. Rather, alternative entry point names can be used during the source location search. The files of interest are found in the following data sets:

- *yourHLQ*.CONFIG.DATA(ALIASES)
- *yourHLQ*.SDMHDATA(DMHALIAS)

Both of these 80-byte text files provide a mechanism by which you can easily add and maintain lists of your program aliases.

To identify alias programs that should resolve run unit names for your installation, you need to update the user-customized file that is used by WebSphere Studio Asset Analyzer processing, as described in step 2 of "Resolution during postprocessing" on page 39. This list of entry point-alias program name pairs is contained in the ALIASES file. The ALIASES file includes instructions, format rules and usage examples. By default, the ALIASES file has no entry point-alias program entries. There are no wildcard characters; entry point names must be explicit. During subsequent installation of future releases and PTFs for WebSphere Studio Asset Analyzer, the product will not update or overwrite the *yourHLQ*.CONFIG.DATA(ALIASES) file.

A similar list of alias programs, used in combination with the user-customized ALIASES, can be found in DMHALIAS. By default, the DMHALIAS file has no entry point-alias program entries. During subsequent installation of future releases and PTFs for WebSphere Studio Asset Analyzer, the product may update the *yourHLQ*.SDMHDATA(DMHALIAS) file. Therefore, in order to preserve your updates, it is recommended that any modifications be applied only to the ALIASES file.

The list of program names from the ALIASES and DMHALIAS files are concatenated at runtime, with the ALIASES file being in the first position. If duplicate entries are found for the same entry point in both the ALIASES and DMHALIAS files, the first-in, first-out rule applies. That is, the information from the ALIASES file will be analyzed first and therefore, the one in effect during postprocessing.

As an example, when a called entry point matches a name in the ALIASES or DMHALIAS files, WebSphere Studio Asset Analyzer searches for source code using the alias name instead of the original entry point name. The Run unit details page indicates that the resolving entry point is an alias, with the **Resolve type** column for the entry point displaying `Resolved by Alias`, and hyperlinks are presented for the Program and Source location columns providing the source for the alias entry point is located in the inventory.

You should review and update the ALIASES file to include known aliases that users are most likely to encounter. You can update ALIASES (and DMHALIAS) at any time. There is no compile or rebind required for WebSphere Studio Asset Analyzer. The next time postprocessing runs, the modified ALIASES and DMHALIAS will be in effect.

RELATED TASK HELPS

Taking inventory > Refining your inventory > Defining alias names for programs

RELATED ONLINE HELP

Help for product pages > Explore MVS assets > Details pages > Run unit

Part 3. Usage practices

Chapter 8. Searching for Assets

You can use either of the following two search capabilities to locate an asset or set of assets of interest:

Simple search
Based on matching asset names.

Advanced search
Based on matching a set of filtering criteria.

Use these search features of WebSphere Studio Asset Analyzer to quickly focus your scope of a large volume of data to something much more manageable for you to explore.

When you perform a simple name search from the Home page, the results appear on the Home page and include all asset types. Each asset displays a **Total** column that indicates inventory count and a **Results** column that indicates the resulting number of matches for your search. However, searching over many assets (such as from the Home, Explore MVS assets, or Explore distributed assets pages) is slower when you use broader criteria and faster when you do the same search using more specific filtering criteria.

For example, looking for `A*` rather than `ACCT-NUM-*` is more resource intensive. It is always best to provide the most specific search criteria that are practical. This is especially true for larger-volume inventory data, such as data elements, literals, DB2 columns, and DD names.

When you use a pattern-matching name search over multiple types of assets (such as from the Explore MVS assets or Explore distributed assets pages), the search is slower due to the number of queries that must be run. The number of required queries is greatly reduced within the context of a specific asset type. When you are interested in a specific asset type, open the summary page for that asset type, and use the basic **Search** (by asset name) or **Advanced search** (by asset name plus various other criteria). The criteria that are available for **Advanced search** varies according to the asset type.

RELATED ONLINE HELP
Help for the product pages > Explore MVS assets > Actions > Search

Search examples

For examples that demonstrate the rules that are used for basic name searches, see the online help: **Analyzing assets > Searching > General search rules**

RELATED ONLINE HELP
Index > search > rules for typing search strings

Interface differences

You should be aware of the following interface differences between MVS and distributed searches:

MVS

- Name searches of MVS assets are upper case by default. Click **Type mixed case** to use mixed case.
- The **Advanced search** criteria are cleared when you close the section by clicking the **x** icon.

Distributed

- Name searches of distributed assets are case-sensitive by default. Click **Ignore case** to have case ignored.
- Advanced search criteria are retained when you close the section by clicking the **x** icon.
- There are many **Advanced search** attributes from which to choose. Use the Path attribute to distinguish between several copies of the same asset. Most distributed assets have a Path attribute, which points to the physical location from which they were scanned. When trying to identify a specific asset, such as a Bytecode class, from several scanned copies of the same asset, use the Advanced search form to add the Path column to the summary page.
- Results are ANDed across different attributes.
- Results are ORed across the same attribute.
- You can group the results by a particular attribute by clicking **Perform group-by query**.
- You can sort by many of the columns in the results.

You can click **Search assets** on the help topic for any distributed summary page for the distributed ″Search assets″ help topic. You can click **Search** on the help topic for any MVS summary page for the MVS ″Search″ help topic.

RELATED ONLINE HELP
Analyzing assets > Searching > Searching MVS assets
Analyzing assets > Searching > Searching distributed assets

Chapter 9. Impact analysis

Impact analysis identifies the ripple effect of a proposed change to your enterprise assets. WebSphere Studio Asset Analyzer provides an on-demand function for understanding and tracing the scope of a change for a large set of MVS and distributed asset types.

RELATED ONLINE HELP
Analyzing assets > Analyzing the impact of a change

Performing impact analysis

You can initiate an impact analysis in the following three ways:

- Selecting assets from a summary page, and then clicking **Analyze impact**.
- Selecting the **Analyze impact of a change** action on an asset details page.
- Selecting the **Analyze impact for a range of statements** action on Program and certain File details pages.

Each of these impact analysis actions opens a wizard in a separate window, showing the assets that you selected. The wizard has the following options that you should understand:

- You can limit the scope of an analysis to the assets of one or more applications. The analysis processing and results report the impacted assets within the scope of the applications that you select, and also the assets impacted just outside the scope. Further processing of the out-of-scope assets is limited.
- Impact analysis **Level** is a count of iterations at different depths of analysis. **Level** provides a quantitative way for you to limit the iterations of looking for impacted assets while still generating results. **Level** works in relation to how your inventory is structured in conjunction with the impact analysis implementation.
- The **Interrupt analysis** and **Cancel analysis** buttons provide a mechanism for you to interrupt or stop an in-progress impact analysis. The interruption or hard stop might not be immediate, because the process has to get to a logical termination point. Actual results might or might not be recorded, depending on how close to the end of processing the interruption was invoked (that is, whether rows have been committed to the database). If impact analysis appears to be processing for an extended period of time, you should modify the **Level** from **unlimited** to **5**. If the initial reduction to **5** continues to produce long-running analysis, reduce the **Level** to **2**.
- You should follow a scheme for your impact analysis names, which makes it easier to locate impact analyses of interest. The browser interface provides a simple search by impact analysis name only; **Advanced search** provides additional filtering by Site and Application. This is in addition to the **View my impact analyses** and **View all impact analyses** views. It is useful to design the impact analysis name for better organization on the summary page, even if you use one of these views. For example, because the impact analyses are presented in alphabetical order, using ″A... ″ for the impact analysis name results in the impact analysis being presented at the top of the results list. Consider using an application or other acronym or abbreviation at the beginning of the impact analysis name to better group related impact analyses.

You can use the following views to see impact analyses with the indicated sorting:

- **Impact analysis > View my impact analyses**: Opens Impact analysis summary page limited to the impact analyses for your user ID, presented in alphabetical order by impact analysis name.
- **Impact analysis > View all impact analyses** or **Home > Impact analysis**: Opens Impact analysis summary page with all impact analyses in the inventory, presented in alphabetical order by impact analysis name, and sorted by user ID within each group.

Default name: The default name that is generated for an impact analysis can be cryptic. You should review the name that is provided in the Impact analysis wizard and modify it to something meaningful. It is important to consider a naming scheme if you are going to aggregate individual impact analyses.

Default description: The default value for the impact analysis Description defaults to ″GENERATED ...″. For example:

```
GENERATED for Program program_name, Data element data_element_name
```

The text becomes more verbose when multiple assets are included. For example:

```
GENERATED for Program program_name, Data element data_element_name;
program_name:data_element_name;program_name:data_element_name;
program_name:data_element_name;program_name:data_element_name
```

This description can be very helpful for identifying the impact analysis starting points (that is, seeds) at a glance. However, you should review the default description that is provided in the Impact analysis wizard and modify it if appropriate. Here too, it is important to consider a description scheme or template if individual impact analyses will be aggregated.

RELATED TASK HELPS

Determining the impact of changes > Performing impact analysis > Initiating impact analysis

RELATED ONLINE HELP

Analyzing assets > Analyzing the impact of a change > Performing a new analysis
Help for product pages > Impact analysis > Actions > Analyze impact

Initial considerations

To have the most accurate impact analysis, make a complete inventory of your assets and correct as many analysis errors as possible.

Invalid analyses: Impact analyses can be made invalid when you delete or rescan existing inventory, or update or refresh user-defined relationships. Rescanning inventory can cause changes significant enough to make existing impact analyses unusable (for example, if the starting point is redefined or made obsolete). In these cases, and in cases where some part of an analysis is made obsolete after an inventory change, a message appears on the Impact analysis details page that indicates that the impact analysis is ″stale″ or the starting point is ″unknown″.

Running time: Some types of impact analysis starting points might take more time to run than others. How long an analysis takes depends on the structure and architecture of the assets. You can run a single impact analysis for multiple assets of a particular asset type. From a summary page of that type, use the check boxes to select multiple assets, and then click **Analyze Impact**. However, impact analysis of multiple assets results in longer processing time and in a larger results set, compared to running a single impact analysis for each individual asset. The larger results set also requires more time to display. A mechanism to select all assets on a summary page for inclusion in an impact analysis is provided, but avoid using this option for large lists of assets, especially if the impact analysis of some subsets of the selected assets is known to be extensive.

Depending on the architecture of your inventory, you may encounter complex impact analyses that are the result of redundant analysis of assets within programs used like utility functions in your installation. If you suspect this to be the case, you may be able to reduce the processing time and results of these impact analyses by identifying the ″culprit″ programs and processing them as utility programs. As you create or update your inventory (or part of it), identify these routines as utilities. This should help you move toward more manageable and useful impact analyses. However, be aware that there are drawbacks to flagging these programs as utilities: primarily that exploration and analyses of the internal assets, control flow, and data flow through utility programs is not available. There is a trade-off that the administrator needs to evaluate before considering how to treat high-use utility types of functions. See “Identifying utility programs” on page 39 for more information.

RELATED TASK HELPS
Taking inventory > Refining your inventory > Defining utility programs
Determining the impact of changes >
Getting complete results from an impact analysis

RELATED ONLINE HELP
Analyzing assets > Analyzing the impact of a change >
Limiting concurrent impact analyses

Impact analysis in the scope of an application

Using an application to scope an impact analysis can reduce running time by limiting how far the impact analysis goes. However, it can also add processing overhead in that the impact analysis will check each impacted asset as it is traversed to determine if it is part of the application. This can add significant impact analysis processing if multiple applications are selected for the scope of an impact analysis, or if the application selected as the scope has a large number of files and/or transactions.

Generally speaking, you should consider creating applications to serve specifically as scopes for impact analyses. Files and transactions can be included in multiple applications, so adding them to specialized applications will not affect their use with other applications. Use these impact analysis-specific applications as boundaries for impact analyses by selecting them as an impact analysis’ scope. By identifying a specific purpose for your various impact analyses you may be able to also identify some boundaries within which impact analysis will be meaningful or applicable. Your impact analyses can then be run, bounded by the scopes you have designated as useful.

As an example: 1) you want to know which assets, ″owned″ by a particular department, are impacted by changes to some subset of DB2 columns, and 2) you want to limit your results to only those assets in files the department controls. In this case, you could create an application and assign the department’s files to it (program files, included source files, batch JCL, WSDL files, HTML files, Java classes, and so on). When you create or reanalyze an impact analysis for one of the DB2 columns, select the department's application as the scope of the impact analysis. When you look at the details of the impact analysis, they will list the impacted assets within the scope of the analysis, (for example, those in files that are owned by the department), and those just outside the boundary of the analysis or department's ownership. The impact analysis stops at the assets just outside the scope of the analysis.

Select an application scope for an impact analysis only when:

- you are not interested in impacts outside the scope of a specific application or some specific set of applications and
- you are relatively confident that the impact will go beyond the scope of the starting point's application (if the starting point is included in an application) or
- you think it will report impacts on assets in applications that do not include the starting point.

Avoid the use of an application as the scope of an impact analysis if the application includes a majority of the files or transactions in your inventory.

Impacted assets just outside the scope of the analysis are recorded in the impact analysis results, but they are not investigated any further beyond the boundary of the application. You can further investigate impacted assets outside the scope of an application by performing additional impact analysis on those assets.

RELATED TASK HELPS

Using applications > Understanding applications

RELATED ONLINE HELP

Help for product pages > Home > Common actions > Create a new application

Avoiding long analyses

You should avoid the following types of analyses due to the extremely long running time:

- Statement ranges in which many data elements are defined or referenced: avoid a statement range with an unreasonable number of data elements. For example, a range of fifty lines that contains fifteen data elements is more reasonable than a range of fifty lines that contains 100 data elements. If you are unsure how many elements are in a range, do the following steps to determine the number of elements:
  1. Set **Level** to 1 in the wizard, and then analyze.
  2. On the Summary tab of the analysis results, review the count of data elements to determine how many data elements are in the statement range.
  3. Analyze in full if there is a reasonable number of data elements.
- Data elements that use a universal work area or buffer
- Analysis of elements that are pointer types

RELATED ONLINE HELP

Help for product pages > Impact analysis > Wizard > Following progress

Dealing with a large number of data elements

You can use the following alternatives to running an impact analysis on a large number of data elements:

- If the data elements that you want to analyze are defined together in included source or in a program, you can select the included source or program, and then choose a range of statements that include the elements that you want to analyze.

  This kind of analysis combines the impact results more efficiently and detects duplicate impacts better than selecting each element and running an independent analysis on each of them. However, there is no way to exclude elements from the range of statements that you select. Therefore, this method of selection is not practicable if the data element declarations are widespread and interspersed with other elements.
- If the data elements that you want to analyze are all associated with a smaller number of DB2 columns, you can select the DB2 columns for impact analysis.

  This method of implicitly selecting the data elements also combines the impact results more efficiently and detects duplicate impacts better than selecting each data element and running independent analysis on each of them.
- Break large sets of assets into subsets.

  You can select subsets by choosing assets that are listed together in another asset's details. For example, in exploring a program, you can list all of its entry points, data stores, or data elements. Select subsets of assets from these lists for analysis.

  You can also select subsets by searching for and listing assets with more qualified search criteria, for example: all the ″used″ data elements with name RPT* that are character type with a physical length of 200. Consider including the selection criteria in the impact analysis **Description** or **Name** to differentiate the impact analyses on an Impact Analysis summary page.

RELATED TASK HELPS
Determining the impact of changes > Performing multiple impact analyses

Aggregating analyses

If you break your impact analyses into subsets, you can combine a set of results by defining a custom query. The results can be qualified by impact analysis names and can be used to show all impacted assets of a particular asset type or all the starting points in a set of impact analyses.

You can modify the following sample queries to suit your needs. If you want to aggregate a set of impact analyses that have a common description, then update the SQL to accommodate a common **Description** text rather than using a common template for the impact analysis name.

Show all impacted programs

Use a custom query similar to the following query when you have named all your impact analyses with a common naming scheme, for example, ′SomeName*′ where ′*′ differentiates the individual impact analyses. To avoid duplicate entries for resulting programs, neither the impact analysis name nor project_id should be displayed in the results.

The following list offers guidance on setting values for selected options on the Custom query wizard. Review the information at the top of each wizard page for other options.

Set core SQL query

```
select distinct
   ctype.name as cmpntType, ci.cmpnt_type_id
 , cu.comp_unit_name, ci.related_id_1
 from ${tschema}.DMH_CHANGE_PROJECT cp
      , ${tschema}.DMH_CHANGE_IMPACT ci
      , ${tschema}.DMH_CMPNT_TYPE ctype
      , ${tschema}.dmh_compile_unit cu
 where  cp.name like 'SomeName%'
   and ci.cmpnt_type_id = 1
   and cp.project_id = ci.project_id
   and ci.cmpnt_type_id = ctype.cmpnt_type_id
   and cu.comp_unit_id = ci.related_id_1
```

Column information

- For better readability, provide labels for the columns: like `Asset type` for cmpntType and `Program name` for COMP_UNIT_NAME. You can update other labels as appropriate.
- If you want to be able to search the results for particular program names, select the **Search column** and **Sort column** check boxes for the **COMP_UNIT_NAME** column.
- Select the **View column** check box for all columns.
- Review these options to verify that you are selecting what you want to see.

Search column options

Note: On this and subsequent pages of the Custom query wizard, column names are referred to using the labels you assigned on the **Column information** page. The following guidelines use `Program name` for the COMP_UNIT_NAME column.

Select the **Simple search column** check box for the **Program name** column and specify `*` for the **Default search string.**

Default view columns

Add **Program name** to the **Current** list under **Default view columns.**

View column options

Select the **Has hyperlink** check box for the **Program name** column.

View column value hyperlinks

Type the following for the URL for the **Program name** column:

```
$C{Urls.WsaaServlet}?pagetype=details&cmpntType=$P{CMPNT_TYPE_ID}&key=$P{RELATED_ID_1}
```

When the Custom query wizard finishes, you should be able to select the query name that you specified and see all the programs in any impact analyses named ’SomeName%’, or whatever you indicated.

Show all impacted data elements and programs

The following custom query is written for impact analyses with a common naming schema, like ’SomeIAName*’, where ’*’ differentiates the individual impact analyses. To avoid duplicate entries for resulting data elements, neither the impact analysis name nor project_id can be displayed in the results.

The following list offers guidance on setting values for selected options on the Custom query wizard. Review the information at the top of each wizard page for other options.

Set core SQL query

```
select distinct
     ctype.name as cmpntType, ci.cmpnt_type_id
   , cu.comp_unit_name, dea.comp_unit_id
   , sym.symbol_text, dea.alias_id
   , mem.member_name, dea.member_id
   , lib.library_name, lib.library_id
   , mem.language_cd, mem.member_type_cd
   , dea.level, dea.data_type_cd, dea.phys_offset
   , dea.logical_length, dea.phys_length
   , dea.scale_amount, dea.parent_alias_id
   , dea.file_line_no, dea.column_no as member_column
   , dea.decl_start_line
   , libCU.library_name as pgm_library_name
   , libCU.library_id as pgm_library_id
   , memCU.member_name as pgm_member_name
   , memCU.member_id  as pgm_member_id
   , site.name as site_name, site.site_id
from ${tschema}.DMH_CHANGE_PROJECT cp
      , ${tschema}.DMH_CHANGE_IMPACT ci
      , ${tschema}.DMH_CMPNT_TYPE ctype
      , ${tschema}.dmh_symbol sym
      , ${tschema}.dmh_data_elmt_alas dea
      , ${tschema}.dmh_member mem
      , ${tschema}.dmh_library lib
      , ${tschema}.dmh_compile_unit cu
      , ${tschema}.dmh_site site
      , ${tschema}.dmh_member memCU
      , ${tschema}.dmh_library libCU
where cp.name like 'SomeIAName%'
   and ci.cmpnt_type_id = 2
   and cp.project_id = ci.project_id
   and ci.cmpnt_type_id = ctype.cmpnt_type_id
   and dea.comp_unit_id = ci.related_id_1
   and dea.alias_id = ci.related_id_2
   and dea.symbol_id = sym.symbol_id
   and dea.member_id = mem.member_id
   and mem.library_id = lib.library_id
   and dea.comp_unit_id = cu.comp_unit_id
   and lib.site_id = site.site_id
   and cu.member_id = memCU.member_id
   and memCU.library_id = libCU.library_id
```

Column information

- For better readability, provide names for the columns: like `Asset type` for cmpntType, `Program name` for COMP_UNIT_NAME, and **Data Element name** for SYMBOL_TEXT. You can update other labels as appropriate.
- If you want to be able to search the results for particular data element and program names, select the **Search column** and **Sort column** check boxes for the **SYMBOL_TEXT** and **COMP_UNIT_NAME** columns.
- Select the **View column** check box for all columns.
- Review these options to verify that you are selecting what you want to see.

Search column options

Note: On this and subsequent pages of the Custom query wizard, column names are referred to using the labels you assigned on the **Column information** page. The following guidelines use `Asset type` for the cmpntType column, `Program name` for the COMP_UNIT_NAME column, and `Data element name` for the SYMBOL_TEXT column.

Select the **Simple search column** check box for at least the **Program name** and **Data element name** columns and specify * for the **Default search string.**

Default view columns
Add **Program name** and **Data element name** to the **Current** list under **Default view columns.**

View column options
Select the **Has hyperlink** check box for the **Program name** and **Data element name** columns.

View column value hyperlinks

- Enter the following for the URL for the **Program name** column:

```
$C{Urls.WsaaServlet}?pagetype=details&cmpntType=1&key=$P{COMP_UNIT_ID}
```

- Enter the following for the URL for the **Data element name** column:

```
$C{Urls.WsaaServlet}?pagetype=details&cmpntType=$P{CMPNT_TYPE_ID}&key=$P{COMP_UNIT_ID}++_$P{ALIAS_ID}
```

RELATED TASK HELPS
Determining the impact of changes >
Aggregating results from multiple analyses

RELATED ONLINE HELP
Help for product pages > Explore Custom queries > Custom query wizard

Adjusting analyses

As an impact analysis of multiple assets is running, intermediate ″snap shots″ of the analysis are committed. As impact information is committed to the database, it becomes available for viewing. From a browser window that is displaying a running impact analysis of multiple starting point assets, you can refresh the view to see the growing set of impacted assets. An impact analysis commit point is hit after each starting point is completely analyzed. When the commit point occurs, if there are at least 100 impacts pending, then they are saved. An unconditional commit point occurs after the last starting point is completely analyzed to commit the last set of the identified impacts. The more complex the impact analysis of a starting point is, the more impacts that are created.

You can alter the database commit threshold of 100 impacts, if you want to raise the likelihood of a commit happening between starting points and have the ability to see the incremental updates to the analysis before the impact analysis completes. To alter the commit threshold, update the wsaa-impact-asset.xml configuration file with a smaller value for the <dbCommitThreshold> parameter. This value should be in the range of 1 to 999, where the smaller the number, the more often a commit occurs. Restart the WebSphere Studio Asset Analyzer application to pick up the modified parameter.

If an impact analysis with multiple starting points runs for a very long time, you can investigate each of the starting points to determine if any particular asset is causing the extended processing. When you use the Impact analysis wizard to create or reanalyze an analysis, you can select one of the seeds and view it. From the selected asset's details page, you can select the action to analyze its impact. If you find that one particular asset is taking a long time to analyze, you can investigate it and its impacts outside the scope of the larger impact analysis.

You cannot remove a starting point from an impact analysis once it is created. To remove the asset, you should delete the particular impact analysis, and then create a new impact analysis, selecting the original set of assets excluding any in which you are no longer interested.

Using impact analysis for field expansion projects

Field expansion, the process of changing the size of a data definition, is a common scenario in many WebSphere Studio Asset Analyzer installations. The Y2K effort is a typical, and perhaps the most well-understood, example of a field expansion effort. To successfully expand a field, you need to do the following work:

1. Identify all uses of the data definition that is to be expanded. The result of this identification (in WebSphere Studio Asset Analyzer terminology) is a set of *seeds* that you can then use as input to impact analysis.
2. Given a list of seeds, analyze the impact of changing each of those fields throughout the enterprise. The result is a set of assets that are identified as impacted.

Review the following two-step approach if you anticipate using impact analysis for a large or enterprise-wide field expansion project:

1. WebSphere Studio Asset Analyzer gathers metadata about both the definition of a data element and its uses. The relationship between the definition and the uses is one-to-many, meaning typically there are several uses for a single definition. The identifier that is associated with the definition is stored as a symbol (in the DMH_SYMBOL table). The relationship between that symbol and the element instances based on that data definition are maintained as data elements (in the DMH_DATA_ELMNT_ALAS table).

   Take, for example, an invoice number. The definitions of data for an invoice number may be represented across your enterprise using a variety of different names: INV, INVNUM, INVOICE, and so on. To find the complete definition, you would search for any data element with a name like INV*. The resulting list of data elements, in a variety of different programs, can then serve as the basis for impact analysis of the data definition in the next step.

   To do this search, you have to understand the set of names for which you might be looking. You may further qualify your search for data elements using a data type, length, and other data properties.

   Many organizations find it difficult to accomplish this first step, finding the data elements across their programs that should serve as the seeds of the overall data definition impact analysis, unless they have diligently followed strict naming conventions and other programming standards. While it is unlikely that performing this first step involves impact analysis, WebSphere Studio Asset Analyzer can help with this step in other ways. It has the set of metadata in place (provided that you have completed the inventory of all your assets) to help identify and explore the uses of the data definition that are to be expanded.
2. Once you have identified the data elements in their various programs that make up a data definition, you can use them as starting points for the impact analyses that identify the assets in the WebSphere Studio Asset Analyzer inventory that will be impacted by a change to the data definition. Impact analysis of multiple data elements for a definition is necessary because performing impact analysis on a single data element in a program may not

provide all impacted assets for the data definition. Impact analysis for a data element is performed within the context of its program. Depending on how your data is moved into and out of that particular data element, WebSphere Studio Asset Analyzer may or may not traverse all the relationships that emanate in all directions for the data definition in general. It may require analysis of multiple data elements in one or many programs in order to gather the complete impact analysis of the overall data definition.

There are two approaches to analyzing the data elements that make up a data definition. You can analyze them all together in one impact analysis or you can perform analysis on each one. In most cases, and especially when a data definition is widely defined or used, tracing everything all at once is not necessarily the most useful approach. It can be time and resource consuming while it is processing and it can also be cumbersome at display time due to its results being relatively large. Therefore, in most cases, perform impact analysis on each (and all) of the data elements that constitute a data definition.

Once impact analyses have been performed you can analyze your results. You might aggregate your results, using custom queries, to get the complete impact analysis ″picture″ across a data definition. For example you may want to summarize all the impacted programs, DB2 tables, files, transactions, and so on, that are impacted by a change to the data definition (data elements). You may also want to look through the various results to find data elements that are impacted by a data definition change, not because they are a named part of the data definition itself, but because they are related to it through program code that references or moves data in a field to another structure that is not named predictably. You might find that some structure or element may need to be rolled into the set of data elements that are the data definition, despite its unconventional name.

Depending on the size of your enterprise, this second step is not something that will happen quickly, but it is doable and it is efficient compared with doing this type of analysis manually. WebSphere Studio Asset Analyzer assists with this kind of scenario by providing the following capabilities:

- The selection of single or multiple seeds for analysis
- Control and monitoring of the progress of any analysis
- Visibility of interim results while an analysis of multiple assets is running
- Interruption of long-running analyses that might need to be broken into subsets
- Custom queries that aggregate the individual impact analysis results to produce specialized reports of data definition impacts across the enterprise inventory
- Specialized naming and describing of analyses results in order to group the analyses for data definition analysis

Chapter 10. Navigating diagrams

WebSphere Studio Asset Analyzer uses scalable vector graphics (SVG) diagrams to display the relationships among assets. For example, the following diagrams are available for MVS assets:

- Component diagrams:
  - Application
  - Transaction flow
  - Batch job
  - CICS group
  - CICS transaction
  - Impact analysis
  - IMS transaction
  - Program
  - Run unit
- Structure diagrams for COBOL and PL/I programs
- Control flow diagram for COBOL programs

Ensure that Adobe's SVG plugin is installed on your client, otherwise the diagrams will not render. [1] If the SVG plugin is not installed, a message appears where the diagram should be drawn. The message provides a link for automatically installing the plugin and also indicates that both Run ActiveX and Script ActiveX options must be enabled in Internet Explorer. If these settings are not enabled, the product will continue to behave as if the SVG plugin is not installed. A user needs external Internet access to use the link for Adobe's Web site.

WebSphere Studio Asset Analyzer provides several options for working with larger, complex diagrams. For example, you can use the following techniques to navigate such diagrams:

- Use **Zoom in** and **Zoom out** to view parts of a large diagram. Scroll bars appear around the diagram when necessary.
- Use marquee selection to zoom into a region of the SVG diagram: press the Ctrl key while you draw a selection rectangle over some portion of the diagram. Use Shift+Ctrl to zoom out.
- To pan around the diagram, press the Alt key and drag the diagram.
- For Batch job diagrams, click **Filter** on the diagram page to expose the following filtering criteria to simplify the diagram display:
  - **Steps to show** (select and deselect)
  - **Hide special JOB DDs** (such as JOBLIB, JCLLIB, JOBCAT)
  - **Hide special step DDs** (such as STEPLIB, SYSMDUMP, SYSUDUMP)
  - **Hide DDs not using a data set** (such as SYSOUT)

  When you use any of the these filtering criteria, you can then navigate through the diagram by using the Previous step and Next step arrows.

[1] At the time of writing this document, no other browser has sufficient support for SVG, and the Adobe plugin does not support other browsers. If this is a problem for you, you might want to check with IBM customer support for the latest status.

If you forget the shortcut keys, click **Help** on the SVG diagram popup menu to view them.

If you have the diagram zoomed out so far that you cannot see the details of a given node or line in the diagram, move your mouse pointer over either to see the associated tooltip.

As shipped, the product does not display scroll bars for SVG diagrams, even when a diagram is zoomed to the point where the content overflows its space on a page. While you can still zoom and pan the diagrams using links, keyboard and mouse navigation, and the SVG viewer's pop-up menu, the diagrams are much easier to navigate if the scroll bars are present. To enable scroll bars on your system, download and install the SVG scroll bar package using the instructions in the technote Enabling scroll bars in WebSphere Studio Asset Analyzer SVG diagrams.

RELATED ONLINE HELP

Help for product pages > Explore MVS assets > Component diagrams

RELATED TECHNOTE

Enabling scroll bars in WebSphere Studio Asset Analyzer SVG diagrams

Chapter 11. Creating and running custom queries

A custom query is a wrapper around a tailored SQL select statement. Custom queries display as user-defined summary pages that extend the functionality of the interface to the WebSphere Studio Asset Analyzer database. Custom queries enable you to use the flexibility of summary views (that is, sorting, searching, paging, grouping, exporting) against a custom SQL SELECT statement to quickly gain access to valuable metadata for unique reporting requirements.

Custom queries have the following characteristics:

- Are created and edited in a browser interface wizard.
- Provide read-only SELECT access to the data. (UPDATE and DELETE are not allowed so that metadata integrity is maintained.)
- Can by run by one of the following mechanisms to give a specialized report:
  - By selection from the Custom query summary page.
  - As a scoped user action from most pages of the browser interface, when you assign the query to a page.

RELATED ONLINE HELP
Help for product pages > Explore custom queries

Tasks by user class

You run the Custom query wizard to create and update queries. You can perform certain tasks only if you are an administrator.

The following tasks can be performed by all users:

Create a custom query by doing one of the follow sets of steps:

1. From the summary page:
   a. Click **Explore > Custom queries**.
   b. Click **Create new custom query** in the **Actions** list.
2. From the details page:
   a. Click **Explore > Custom queries**.
   b. Click a custom query.
   c. Click **Copy this custom query** in the **Actions** list to create a new custom query based on the existing custom query.

Update a custom query by doing the following steps:

1. Click **Explore > Custom queries**.
2. Click a custom query.
3. Click **Edit this custom query** in the **Actions** list.

Run a custom query by doing one of the following steps:

- For a custom query that is assigned to a page, click the name of the query in the **Actions** list for that page.
- For a custom query that is not assigned to a page, click the query on the Custom query summary page.

Only administrators can perform the following tasks:

Assign the custom query to a different user

An administrator can change the user ID to which the query is assigned. You change the ID by updating the query and changing the **User** field in the wizard. This field is read-only for non-administrators.

Delete queries

An administrator can delete queries from the database by selecting queries and clicking **Delete** on the Custom queries summary page. These controls are not enabled for non-administrators.

Gaining knowledge and experience

Only users who are well versed in SQL and knowledgeable of the WebSphere Studio Asset Analyzer data model should employ custom queries.

SQL: Constructing custom queries requires a significant knowledge of SQL, and, depending on the complexity of the request, might demand a skill level beyond the understanding of basic SQL syntax. The wizard provides SQL validation, however, well-formed SQL is required for accuracy and performance. If the query is poorly written, a timeout on the data source could occur, depending on the WebSphere Application Server setting. Consider identifying a small group of SQL-experienced users who can serve as the resource for composing custom queries for your larger user community.

Data model: Observe the following best practices to gain a better understanding of the WebSphere Studio Asset Analyzer data model:

- Compose a purposeful and specific question, so that you clearly define the result set that you want.
- Identify the top-level assets that are involved: are they programs, include members, data elements, transactions, batch jobs, job steps, data stores, data sets?
- Become knowledgeable about the metadata that is stored in the database with respect to the assets of interest:
  - Explore the browser interface to identify what asset information is already available on explore, summary, details, and diagram pages.
  - Examine the queries that are used by WebSphere Studio Asset Analyzer on a variety of asset pages. You can easily accomplish this by appending `&SHOWSQL=1` to the end of a page address, and then refreshing the page.
  - Review the database documentation in the online help. Look for tables that house the assets, columns with data of interest, foreign key relationships, and so on.
- Start with simple SQL, and then add more complexity with each successful run and greater data model understanding.

The online help provides two examples of creating custom queries that are assigned to asset details pages. You should review and experiment with these as an introduction to custom queries. These examples are also illustrated in Flash demos on the IBM Education Assistant.

RELATED ONLINE HELP

Administrative information > Database tables >
Database tables by submodel
Help for product pages >Explore Custom queries >
Example: Creating a custom query as a run unit action

Help for product pages > Explore Custom queries > Example: Creating a custom query as a program action

RELATED REFERENCES
DB2 Universal Database for OS/390 and z/OS SQL Reference (V7)
DB2 Universal Database for z/OS SQL Reference (V8)
WebSphere Studio Asset Analyzer on the IBM Education Assistant

When to use custom queries

Custom queries are appropriate for the following situations:

- When you need information that is not available through the interface but can be retrieved from the database through a complex SQL query.
- When you have defined custom tables in the database, you can create queries to extend the interface to include the information in those tables.
- When you need unique reports to aggregate summary information.

Custom queries are intended to be used for such situations. You should run one-time queries by using **Database > Execute a query** rather than creating a custom query.

Establishing naming conventions

Custom query names are free-form text less than 129 characters in length. You should keep the following practices in mind when naming your queries:

- The Custom queries summary page does not automatically filter the custom queries. By default the page displays all known custom queries in the inventory, sorted by query name. Consider establishing naming conventions to give meaningful, task-oriented names for easy identification by users. You can then use the simple and advanced search functions on this page to filter the results.

  When the number of queries exceeds the default group size of displaying results on a page (15), use the **Perform group by query** option of advanced search. This option groups the results by select attributes, including **Create user**. When you group the results by the user ID that created the query, the results show all IDs with the corresponding counts of queries, which are links. Click the count to display just the queries for that ID. Using such filtering avoids having to page through the results or adjust the **Show groups of** setting.
- Custom queries that are assigned to explore, summary, or details pages are added to the existing **Actions** list in alphabetical order. Here too, it is important to consider meaningful, task-oriented names for easy identification by users. In addition, consider a short name, because the custom query name is the text string that is displayed in the list.

Choosing the expiration method

When you create a custom query, you choose the method by which the query expires by selecting a method from the **Expire method** list of the Custom query wizard. The method is used to determine if and when a custom query can be deleted automatically from the inventory. Employ the following best practices when choosing the method:

- By default **Expire method** is set to **Don’t expire**. When selecting a method, take into consideration whether a custom query has limited use: for a short period of time (select **From create date**) or for a longer period of time (select **From**

last-accessed date). When you select **From create date** or **From last-accessed date**, the **Expire** field is enabled, providing a choice of 5, 30, 90, and 365 days.

- Expired custom queries are not automatically deleted from the database. To remove expired custom queries from the database, execute the **Clean up expired** action available on the Custom query summary page. The **Clean up expired** action requires a user ID with administrative authority.

RELATED ONLINE HELP
Help for product pages > Explore Custom queries > Custom query wizard

Verifying that queries remain valid

The **Is valid** column of the Custom query summary page indicates if the corresponding custom query contains valid SQL for the current WebSphere Studio Asset Analyzer database. Possible values are **Yes** (valid), **No** (invalid), or **Unknown**. **No** identifies a custom query that is no longer valid because the underlying database schema has changed. **Unknown** indicates that this custom query has not been executed since a migration has occurred, and therefore, it is not known whether it is valid or not. To revalidate custom queries that are in the Unknown state, perform one of the following:

- In turn, execute the custom queries individually to reestablish a Yes or No value in their **Is valid** column.
- Use Advanced search to identify all custom queries in the Unknown state and execute the **Validate** table action. The **Validate** action requires a user ID with administrative authority. This task executes in the background to reset the **Is valid** column of all submitted custom queries to Yes or No.

You should inspect the **Is valid** column after any database migration or any in-house activity that otherwise modifies the underlying database objects to verify that all custom queries remain valid.

RELATED ONLINE HELP
Administrative information > Database tables

Custom query catalog

To promote the sharing of custom queries that have proven valuable in typical client scenarios, the WebSphere Studio Asset Analyzer Web site includes a Custom Query Catalog at http://www.ibm.com/support/docview.wss?uid=swg24014566. This catalog contains a set of custom queries that have been exported into XML file format for V4.2 and later releases of the product. You must use the version of the catalog that matches the version of WebSphere Studio Asset Analyzer you have installed, as these queries are specific to the schema that goes with each release.

To import custom queries you must be logged on to the WebSphere Studio Asset Analyzer browser using an ID with administrative authority. Follow the instructions in the online help topic **Help for product pages > Explore Custom queries > Exporting and importing** to import queries from the catalog on the Web site by pointing to the URL containing the query you are interested in. Once in the Custom Query Import wizard, you will see a description of the query and can then decide whether or not to continue with the import process. Once the query is in your inventory, you are free to rename it, update the description, and update the content in any way you see fit for your enterprise.

If you would like to contribute to this catalog by providing queries you have created and found useful at your installation, contact your IBM technical support team.

Related technote
WebSphere Studio Asset Analyzer custom query catalog

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 give 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 (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those 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
J46A/G4
555 Bailey Ave.
Santa Teresa Lab
San Jose CA 95141-1003
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 information 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 will be the same on generally available systems. Furthermore, some measurements 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.

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. 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 IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. 2001, 2007. All rights reserved.

Trademarks

AIX
CICS
CICSPlex
ClearCase
COBOL/370
DB2
DB2 Universal Database
IBM
IMS
Language Environment
MVS
MVS/ESA
OS/390
Redbooks
WebSphere
z/OS
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Readers' Comments — We'd Like to Hear from You

IBM WebSphere Studio Asset Analyzer for Multiplatforms Version 5 Release 1
Best Practices Guide

Publication No. SC19-1168-01

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:
- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: 800-426-7773
- Send your comments via e-mail to: www.ibm.com/software/ad/rcf/

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You

SC19-1168-01

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Reader Comments
DTX/E269
555 Bailey Avenue
San Jose, CA
U.S.A.
95141-9989

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line