IBM WebSphere Studio Asset Analyzer for Multiplatforms
Version 4 Release 2

Best Practices Guide

IBM WebSphere Studio Asset Analyzer for Multiplatforms
Version 4 Release 2

Best Practices Guide

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 57.

Second Edition (June 2006)

This edition applies to Version 4 Release 2 of IBM WebSphere Studio Asset Analyzer for Multiplatforms Version 4 Release 2 (Program Number 5655-M22).

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 three sources of product documentation that are available from the WebSphere Studio Asset Analyzer product Home page of your installation of WebSphere Studio Asset Analyzer:

- ″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.
- ″Task help″ refers to the product help topics that are accessible from the **Task Help** list on the product Home page.
- ″Product documentation″ refers to the PDF documents that are available from the PDF documents entry in the online help navigation pane.

In addition there is a WebSphere Studio Asset Analyzer 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.

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.

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 by 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 update custom queries.	• Take an inventory or any inventory-related activity such as creating or assigning concatenation sets, queuing files or containers for analysis. • View summary information about the state of the database. • Delete assets from the database.

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 -> Metrics information • View and act upon Database -> Unreferenced assets • Delete analysis queue entries • Delete applications, bookmarks, custom queries • 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
Notes: 1. Can delete any analysis, regardless of who created it. Non-administrators can delete only their own analyses.

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 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 no administrators. (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 ONLINE HELP
Task Help: **Getting started -> Learning the types of users**

Related tasks

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

Part 2. Administrative practices

Chapter 3. Taking an inventory

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

Container scan
Create a list of files in a container, and gather container attributes metadata

Classification scan
Identify the language and file type, and gather language-neutral metadata (a shallow scan)

Analysis scan
Gather 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 is provided in the *yourHLQ*.SDMHCNTL(DMHJCLBU) and *yourHLQ*.SDMHCNTL(DMHJCLQU) members. It allows you to take an inventory of source assets without access to the browser interface.

CICS: You take an inventory of CICS and IMS 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.

SCM: 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 32 for additional information.

RELATED TASKS
Running DMH0700 while DMH6000 is also running

Best approach

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.

Choosing COBOL and PL/I scanners

WebSphere Studio Asset Analyzer has proprietary scanners for scanning PL/I and COBOL source code if the Enterprise compilers are not available. If your installation is using the prerequisite level of Enterprise compilers, you should use the compiler-based scanners instead of the proprietary scanners. Use Option 0, Set up, of the ISPF panels to specify the compilers for COBOL or PL/I and to verify that the settings that are required for COBOL and PL/I scanning are set correctly.

RELATED TASKS
Chapter 5. Operational Requisites, *Program Directory*
Chapter 3. Defining product defaults, *Configuration and Migration Guide*

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

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

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, the default DB2 system name is NULL, and it is not added to the inventory.
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, 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.V4R2M0.SDMHLOAD,DISP=SHR
//         DD  DSN=CICSVS.TS230.SDFHLOAD,DISP=SHR
//         DD  DSN=CEE.SCEERUN,DISP=SHR
//CESYSIN  DD *
SITE **DEFAULT** Default Site
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 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

RELATED ONLINE HELP

Taking an inventory -> Taking an inventory of MVS assets ->
    Scanning CICS and IMS -> Scanning CICS regions

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 "MVS OR OS/390"
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, each suboption may not be available at your site. For example, the IMS Library Integrity Utilities (LIU) is used to reverse engineer IMS binaries. If LIU is not available at your site, alternatively you can collect inventory for IMS DBDGEN and PSBGEN source files by using the browser interface.

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

RELATED ONLINE HELP
Taking an inventory -> Taking an inventory of MVS assets -> Scanning CICS and IMS -> Scanning IMS subsystems
Taking an inventory -> Taking an inventory of MVS assets -> Scanning CICS and IMS -> Scanning IMS subsystems -> Analyzing IMS transactions.

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*.SDMHCNTL 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*.SDMHDATA 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. No DB2 catalog metadata is dropped on subsequent scans of the catalog, so 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 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)*

RELATED ONLINE HELP
Taking an inventory -> Taking an inventory of MVS assets -> Scanning DB2
Taking an inventory -> Taking an inventory of MVS assets -> Scanning DB2 -> Creating parameters

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 should be in incomplete state following normal completion of the 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 CSD

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*.SDMHDATA(DMHALIAS) 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 DMHALIAS 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 tasks
"Identifying aliases" on page 37

Related online help
Help for product pages -> Database -> Errors
Help for product pages -> Database -> Errors -> Activity log error summary

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. JCL PROCs, Assembler macros, and COBOL copybooks are examples of included source. To best represent your processing environment and inventory, you should locate existing compilation jobs to determine the actual libraries and search order that is 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:

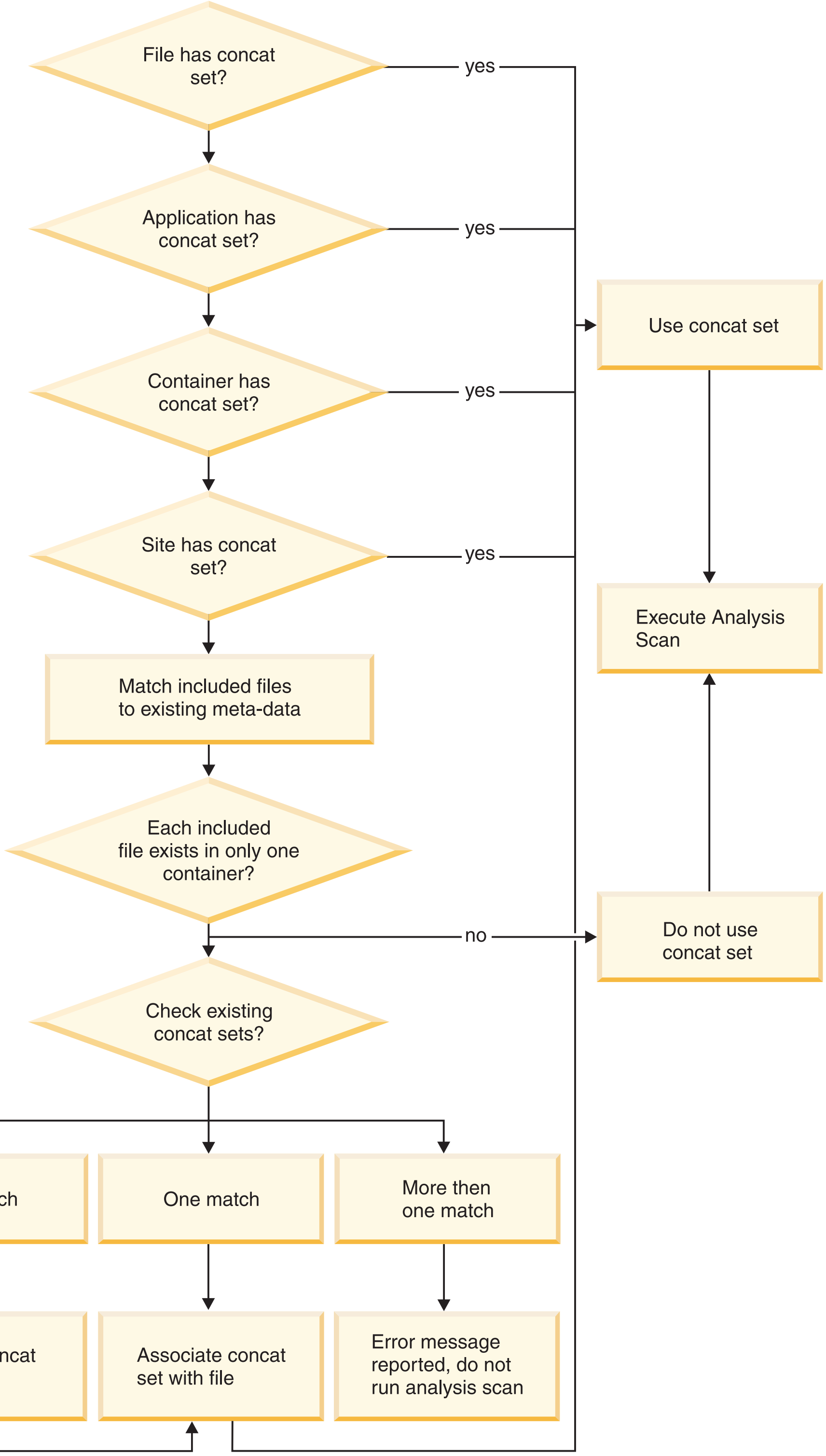
- Uppercase character string of not more than eight characters
- Alphabetic characters, #, @, $, and numeric digits
- The first character cannot be numeric

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 ONLINE HELP

Taking an inventory -> 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.

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 use 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. Configuring the JCL analyzer, *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.

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 ONLINE HELP
Help for product pages -> Home -> Common actions -> Create a new application
Help for product pages -> Home -> Common details pages -> Application
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 rescanned, added or deleted assets. You should consider the following aspect of the snapshot:

- Which snapshot you want: development, unit test, quality assurance, or production.
- If you want more than one snapshot, you could consider defining multiple sites in WebSphere Studio Asset Analyzer. However, you need to be aware that keeping multiple versions of your source by site might encounter performance limitations.
- 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.

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

2 - 98 Subprogram source

99 Recursion encountered

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 Resole 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]

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 multi-level 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.

Options for resolving a missing entry point

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

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

- Include the program name in the *yourHLQ*.SDMHDATA(DMHUTIL) file to resolve the entry point as a utility, and queue it for analysis. This will forgo any further search for program source.
- Include the program name in the *yourHLQ*.SDMHDATA(DMHALIAS) file to resolve the entry point as an alias of another program, if applicable, and queue it for analysis.
- Create a dummy stub program to resolve the primary entry point reference, and submit it for inventory.

Resolution during postprocessing

WebSphere Studio Asset Analyzer resolves run units and builds calling hierarchies during postprocessing 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 (that is, levels 2 - 98), 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, it is flagged as Level 99, and processing continues with the next transfer target.

RELATED TASKS
"Identifying utility programs"
"Identifying aliases" on page 37

Identifying utility programs

To identify utility programs that should resolve run unit names, you need to update the list that is used by WebSphere Studio Asset Analyzer processing, as described in step 1 of the previous section. This list is contained in file DMHUTIL.

DMHUTIL is a text file that contains the names of utility programs and subroutines such as IEBGENER, IDCAMS, and CBLTDLI. It is shipped with a default set of entries from IBM and others, and can be found in the *yourHLQ*.SDMHDATA data set. The DMHUTIL file provides a mechanism by which you can easily add your own and third-party utility programs and subroutines. This information is use by WebSphere Studio Asset Analyzer during postprocessing to decide which called programs should not be candidates for source location when building call hierarchies.

For example, when a called entry point matches a name in the DMHUTIL list, 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 as `Resolved by Utility`, and that no source is linked to it.

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

- The source code for the program is not available and likely will never be available for inventory.
- 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 DMHUTIL file as a wildcard to avoid typing large groups of individual names. For example, the file is shipped with entries for CEE* (Language Environment) and DFH* (CICS).

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

RELATED ONLINE HELP
Help for product pages -> Explore MVS assets -> Details pages -> Run unit

Identifying aliases

To identify aliases that should resolve run unit names, you need to update the list that is used by WebSphere Studio Asset Analyzer processing, as described in step 2 on page 36. This list is contained in file DMHALIAS.

DMHALIAS is a text file that contains entry point-alias pairs that you provide. It can be found in the *yourHLQ*.SDMHDATA data set. The DMHALIAS file provides a mechanism by which you can easily add alias names that will be used by WebSphere Studio Asset Analyzer during postprocessing for run unit resolution.

For example, when a called entry point matches a name in the DMHALIAS file, 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 program is an alias, with the **Resolve type** column for the entry point as `Resolved by Alias`, and having a source location linked to it.

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

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 -> Searching MVS assets -> Search rules examples**

For other search details, see the following online help topics:

RELATED ONLINE HELP
Index -> search -> rules for typing search strings
Analyzing assets -> Searching -> Searching MVS assets -> Searching for multiple words

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.
- 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 TASKS
"Aggregating analyses results" on page 46

RELATED ONLINE HELP
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. The errors can be so destructive as to make the existing impact analysis unusable (for example, the starting point is redefined or obsolete). For MVS assets, a message appears on the Impact analysis details page that indicates that the impact analysis is ″stale″. This mechanism is not available for distributed assets. Instead, the distributed asset details pages include sections for the following analyses:

- Impact analyses that start with this asset
- Impact analyses that show an impact on this asset

If you suspect that an impact analysis might be made invalid because of a rescan, you can go to the asset's details page and see if that particular asset is involved in any impact analyses. If it is, the impact analyses should be recreated or reanalyzed after the rescan.

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 intentionally not provided to avoid extremely large impact analyses.

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 element 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

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.

Aggregating analyses results

If you break your impact analysis 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 data elements, neither the impact analysis name nor project_id should be displayed in the results.

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

1. For better readability, provide names for the columns: type `Asset type` for cmpntType and `Program name` for comp_Unit_name.
2. If you want to be able to search the results for particular program names, mark the **Program name** column to be **Is name**.
3. Mark all columns as **Is visible**.
4. Mark the **Asset type** and **Program name** columns as **Is default**.
5. Mark the **Program name** column as **Has link**.
6. Review the other options to verify that you are selecting what you want to see.

Hyperlinks on columns

1. Type the following address for the URL for the COMP_UNIT_NAME:

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

2. Review the information at the top for other options.

When the Custom query wizard finishes, you should be able to select the query name that you specified and see all the programs in your impact analyses that are named ′D%′, 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.

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.level, 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

1. For better readability, provide names for the columns: type `Asset type` for cmpntType, `Program name` for comp_Unit_name, and **Data Element name** for symbol_text. You can update other labels as appropriate.
2. If you want to be able to search the results for particular program names, mark the **Data element name** or **Program name** column to be **Is name**.
3. Mark all columns **Is visible** or just the ones you will use.

4. Mark at least the **Asset type**, **Program name**, and **Data element name** columns as **Is default**.
5. Mark the **Program name** and **Data element name** columns as **Has link**.
6. Review the information at the top of the page for other options to verify that you are selecting what you want to see.

Hyperlinks on columns

1. Enter the following address for the URL for the COMP_UNIT_NAME:

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

   Enter the following address for the URL for the SYMBOL_TEXT:

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

2. Review the information at the top of the page for other options.

RELATED TASKS
Chapter 11, “Creating and running custom queries,” on page 53

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 in the DMH_SYMBOL table of the WebSphere Studio Asset Analyzer database. The relationship between that symbol definition and the element instances that are based on that definition is also maintained. The data element table (DMH_DATA_ELMNT_ALAS) contains the column SYMBOL_ID that points to a specific symbol.

   For example, say you are interested in Invoice Number, and this is represented across your enterprise by a variety of names: INV, INVNUM, INVOICE. You could search the DMH_SYMBOL table looking for all symbols named INV*, and then use those SYMBOL_IDs to find the associated data elements. The data elements become your seeds for the next step.

   To do this search, you have to understand the set of names for which you might be looking.
2. Once you have identified the data element seeds, you can use impact analysis to perform the analysis. Performing impact analysis on data elements in programs does not provide all impacted assets for any single program and data element instance. Impact analysis for a data element is performed within the context of that program. Depending on how your data is moved into and out of that data element, WebSphere Studio Asset Analyzer might or might not perform the tracing of everything that emanates in all directions from all points in the path.

   The result of tracing absolutely everything is not necessarily the most useful. A better approach is to perform impact analysis on each (and all) of the data elements, and then aggregate your results to get the complete picture. Depending on the size of your enterprise, this is not something that happens 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 multiple seeds for analysis

- The capability to monitor the progress of that analysis
- Visibility of interim results while the analysis is running
- Interruption of the process to avoid long-running analyses that might need to be broken into subsets
- Queries that aggregate the individual impact analysis results to produce a final analysis set

Many organizations find it difficult to accomplish the first step, finding the seeds, unless they have diligently followed strict naming conventions and other programming standards. While it is unlikely that performing this step involves impact analysis, WebSphere Studio Asset Analyzer can help with this step; it has the set of metadata in place (provided that you have completed the inventory of all of your assets) to help identify the uses of the data definition that are to be expanded.

You could start by searching by name for the fields and by performing custom queries on the symbol table, both of which would give you a good starting set of data elements. There might be MOVEs or other similar program code that references or moves data in a field to another structure that is not named predictably. You could use impact analysis to identify such structures.

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.

RELATED ONLINE HELP

Help for product pages -> Explore MVS assets ->
Component 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 time-out 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 accomplished 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.

RELATED ONLINE HELP

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

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.
- In the ExpireTimerTasks section of the /usr/lpp/dmh/config/Common.cfg file is a setting for customQueryCheckIsEnabled. This is disabled by default (that is, set to false). When it is enabled, the timer runs in a background thread to check

for expired queries every *x* milliseconds, where *x* is the value of the customQueryTime setting. When the timer is disabled, expired custom queries are purged only at servlet start-up time. You should enable the timer only if you have many users and very heavy usage of custom queries.

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) and **No** (invalid). **No** identifies a custom query that is no longer valid because the underlying database schema has changed. You should inspect this 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

Notices

References in this document to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe on any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the responsibility of the user.

To obtain current service suggestions and to identify current product service requirements, always contact the IBM Customer Support Center or use S/390 SoftwareXcel to obtain the current ″PSP Bucket″.

IBM may have patents or pending patent applications covering subject matter 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 the

IBM Director of Licensing
IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1099
USA

For online versions of this book, we authorize you to:

- Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy.
- Transfer the original unaltered copy of the documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for payment of any taxes, including personal property taxes, resulting from this authorization.

THERE ARE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine readable documentation.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

CBPDO
CICS
DB2
IBM®
IMS
OS/390
RETAIN
ServerPac
SystemPac
WebSphere
z/OS

ClearCase is trademark of Rational Software Corporation in the United States, other countries, or both.

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.

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

Other company, product and service names may be trademarks or service marks of others.

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

IBM WebSphere Studio Asset Analyzer for Multiplatforms Version 4 Release 2
Best Practices Guide

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

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.

Name

Address

Company or Organization

Phone No.

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

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

Printed in USA