

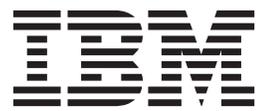
IBM InfoSphere Classic Federation for z/OS
Version 11 Release 1

Installation Guide



IBM InfoSphere Classic Federation for z/OS
Version 11 Release 1

Installation Guide



Note

Before using this information and the product that it supports, read the information in “Notices” on page 37.

This edition applies to InfoSphere Classic Federation for z/OS (program number 5655-IM4) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2003, 2013.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

About this information

These topics provide conceptual, guidance, and reference information for installing InfoSphere® Classic Federation for z/OS®. The topics also describe how to customize the installation environment and install data servers.

This information is available as part of the InfoSphere Classic Federation for z/OS Information Center at <http://pic.dhe.ibm.com/infocenter/iisclzos/v11r1>.

How to send your comments

About this task

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other product information, you can take one of the following actions:

Procedure

- From any topic in the information center at <http://pic.dhe.ibm.com/infocenter/iisclzos/v11r1>, click the **Feedback** link at the bottom of the topic and complete the Feedback form.
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the title, the part number of the title, the version, and, if applicable, the specific location of the text on which you are commenting (for example, a page number in the PDF or a heading in the information center).

Installing Classic federation

Installing Classic federation consists of installing mainframe components and the Classic Data Architect, preparing your installation environment, and customizing the installation to create a functional runtime environment.

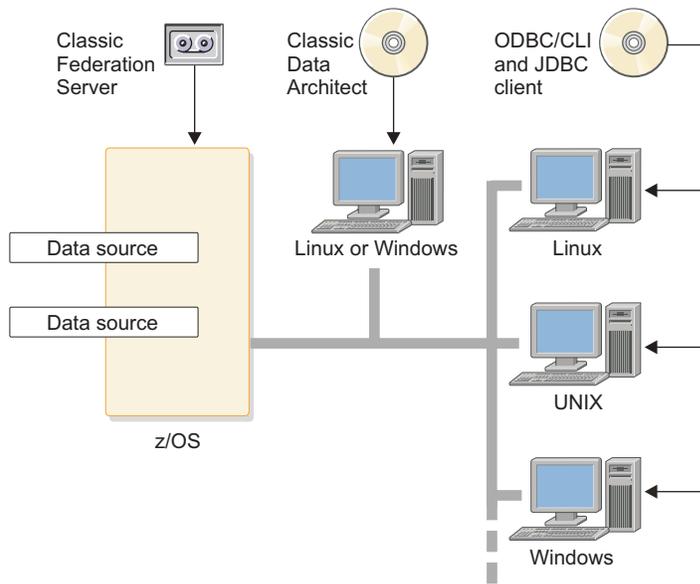
The following table lists the major tasks required for Classic federation installation with a link to where to find information about each task. Perform the tasks in the following recommended order.

Task	Reference
1 Perform the SMP/E tasks to install the components required for Classic federation on the mainframe.	"Installing Classic federation on the mainframe" on page 2
2 Prepare for the installation customization process by completing the tasks in the checklist for setting up the installation environment. Obtaining items such as required authorizations and port numbers will prepare you for the customization procedure.	Setting up the installation environment
3 Customize the server installation environment by completing the tasks in the installation customization process for the type of server that you want to customize.	Customizing the installation environment
4 Install the Classic Data Architect client application to manage server connections and subscriptions, monitor metrics, and perform configuration tasks.	Installing Classic Data Architect
5 Install the ODBC/CLI and JDBC clients	"Installing the ODBC/CLI and JDBC clients" on page 31

System Z Classic federation scenario

IBM® InfoSphere Classic Federation Server for z/OS provides SQL access to System Z data sources, such as DB2®, IMS™, VSAM, Software AG Adabas, CA-Datacom, CA-IDMS, and sequential files.

About this task



System requirements

IBM InfoSphere Classic Federation Server for z/OS contains mainframe components, client components, and it supports a variety of data sources.

To view the hardware and software requirements for InfoSphere Classic Federation Server for z/OS, see <https://ibm.biz/BdD25S>.

Installing Classic federation on the mainframe

The IBM InfoSphere Classic Federation Server for z/OS product is included on tape and the installation instructions are detailed in the product program directory.

About this task

The Program Directory details the system requirements and installation instructions for InfoSphere Classic Federation Server for z/OS.

Setting up the installation environment

After you complete the mainframe SMP/E installation, the next step in the installation process is to set up the installation environment. Setting up the installation environment is a prerequisite to the installation customization process.

The following table provides a checklist of tasks needed to set up the installation environment for Classic data servers.

Table 1. Checklist of installation environment setup for Classic data servers

Task	Reference
Obtain APF library authorizations for the installation load library SCACLOAD	Obtaining library authorizations for the authorized program facility (APF)

Table 1. Checklist of installation environment setup for Classic data servers (continued)

Task	Reference
Assign port numbers for communication for Classic data servers.	Obtaining ports for communication for Classic data servers
Set up resources profiles and security classes for security for Classic data servers.	Securing a Classic data server
Ensure that you have the authorization required to run the Administrative Data Utility (IXCMIAPU). You need this authorization before you run the utility to define the Classic event log and the diagnostic z/OS log streams.	Administrative Data Utility

Customizing the installation environment

The goal of the installation customization process is to simplify the setup of your runtime environment by providing a central place for you to specify the site-specific information that is needed to configure your environment.

The information that you provide is then used as input for generating all JCL and configuration data needed to build the runtime environment.

Installation customization process

Installation customization is a process that allows you to provide setup and configuration information to create a customized installation environment.

The installation customization process involves a set of steps that you perform after you complete the mainframe SMP/E installation. You provide setup and configuration information that is used to generate all of the sample JCL and configuration members in the *USERHLQ.USERXSAMP* data set that require edits. You then run installation customization jobs that are generated based on the parameters that you specify to create a customized installation environment.

The installation customization process is based on the role of the Classic data server. The possible roles for a Classic data server are based on the data sources that you choose to access. You can customize an installation environment for one or more data sources. You specify the role of a Classic data server with the *SERVERROLE* parameter for the installation customization utilities. This parameter controls the installation components that you customize. You create installation data sets (*USERHLQ.USERXSAMP* and *USERHLQ.USERCONF*) that contain the required components for the type of installation that you choose, and you customize only the parameters needed for that environment.

When you complete the installation customization process, an operational environment is established that you can build upon as needed. The environment includes a functional Classic data server and all of the services required for the specified role. All services are pre-configured during the customization process.

Overview of installation customization procedure

The installation customization process consists of the following basic steps:

1. The user samples allocation utility creates a working set of the SCACSAMP and SCACCONF data sets that contain all customized JCL and configuration members. This working set is referred to as the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets.
2. You gather site-specific configuration information needed to customize the environment and enter that information in the customization parameters file.
3. The installation customization utility generates customized JCL and configuration members and stores them in the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets that were created in the first step.
4. You allocate and initialize the following components by using generated customization jobs:
 - z/OS log streams
 - Configuration files
 - A zFS aggregate that you define and format
 - Metadata catalogs
5. You use the generated JCL and configuration members to start the runtime environment.

Installation customization components

The following table lists the components and sample JCL members that you use during the installation customization process.

Table 2. Summary of installation customization components distributed in the SCACSAMP data set.

Component name	Description
User samples allocation utility	Allocates the <i>USERHLQ.USERSAMP</i> and <i>USERHLQ.USERCONF</i> data sets. Populates the <i>USERHLQ.USERSAMP</i> data set with a copy of the customization parameters file (CACCUSP2) and the installation customization utility JCL (CACCUSJ2). The <i>CACINHLQ.SCACSAMP(CACCUSJ1)</i> JCL runs this utility. <i>SCACSAMP(CACCUSJ1)</i> is the JCL that runs the user samples allocation utility. <i>CACCUSJ1</i> is the only member in the distributed SCACSAMP data set that you edit. The JCL comments provide editing instructions.
Installation customization utility	Reads the customization parameters file <i>USERHLQ.USERSAMP(CACCUSP2)</i> and generates the necessary JCL and configuration members in the <i>USERHLQ.USERSAMP</i> and <i>USERHLQ.USERCONF</i> partitioned data sets. The <i>USERHLQ.USERSAMP(CACCUSJ2)</i> JCL runs this utility. <i>USERHLQ.USERSAMP(CACCUSJ2)</i> is the generated JCL that submits the installation customization utility and generates all necessary JCL and configuration members.
Customization parameters file	Contains the installation and customization information that you specify in the form of parameter and value pairs to complete an installation and establish an initial functioning environment. This file is located in <i>USERHLQ.USERSAMP(CACCUSP2)</i> .
<i>USERHLQ.USERSAMP(CACCFSL)</i>	Generated JCL that runs the Administrative Data Utility (IXCMIAPU) to define the z/OS event log stream and a log stream for the diagnostic log for the Classic data server.
<i>USERHLQ.USERSAMP(CACCATFG)</i>	Generated JCL that allocates and initializes the configuration files and the metadata catalog for the Classic data server.
<i>USERHLQ.USERSAMP(CACDS)</i>	Generated JCL to start the Classic data server.

Table 2. Summary of installation customization components distributed in the SCACSAMP data set. (continued)

Component name	Description
<i>USERHLQ.USERSAMP(CECCRZCT)</i>	Generated JCL to define and format a new zFS aggregate to use for a USS file system resident version of the system catalogs. Although you can continue using sequential data set resident system catalogs, zFS file system resident catalogs are recommended.

User samples allocation utility

The user samples allocation utility allocates the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets and populates the *USERHLQ.USERSAMP* data set with a copy of the customization parameters file and the installation customization utility.

The user samples allocation utility performs these functions:

- Allocates the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. These data sets are created with the same characteristics as the distributed SCACSAMP and SCACCONF data sets.
If you run the utility again, the *USERHLQ.USERSAMP* or *USERHLQ.USERCONF* data sets that already exist are reused. The utility replaces the customization parameters file and customization utility JCL members. All other members remain the same.
- Generates the customization parameters file *USERHLQ.USERSAMP(CACCUSP2)* and the installation customization utility JCL *USERHLQ.USERSAMP(CACCUSJ2)*. All input parameters specified for the samples allocation utility are populated in the generated CACCUSP2 and CACCUSJ2 members.

You use the SCACSAMP(CACCUSJ1) job to run the allocation utility. The JCL contains comments with editing instructions. You specify the following input as parameters:

CACINHLQ=CAC.V11R1M00

The value specified for the CACINHLQ keyword must be the high-level qualifier of the installation data sets that the SMP/E installation produces.

CACUSHLQ=USER.V11R1M00

The value specified for the CACUSHLQ keyword must be the high-level qualifier for the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets that the samples allocation utility creates or updates.

CACDUNIT=SYSALLDA

The value specified for the CACDUNIT keyword identifies the disk unit that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter.

CACDVOLM=

The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter.

CACSTGCL=

The value specified for the CACSTGCL keyword identifies the SMS storage class that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter.

CACMGTCCL=

The value specified for the CACMGTCCL keyword identifies the SMS management class that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter.

ISPFHLQ=ISP

The value specified for the ISPFHLQ keyword identifies the high-level qualifier for ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU

The value specified for the ISPFLANG keyword identifies the language prefix for the ISPF installation.

SERVERROLE=(role-name, ...)

The value of the SERVERROLE keyword specifies that the server environment being installed and customized contains the components required for the Classic data server environment. You can specify one or more roles for your Classic data server. If you specify multiple role names, you must separate the names with commas and enclose the names in parentheses.

CF_ADABAS

Specify this value to install the components required for a Adabas access.

CF_DATACOM

Specify this value to install the components required for CA-Datcom access.

CF_DB2

Specify this value to install the components required for DB2 access.

CF_IDMS

Specify this value to install the components required for CA-IDMS access.

CF_IMS

Specify this value to install the components required for IMS access.

CF_SEQ

Specify this value to install the components required for sequential file access.

CF_VSAM

Specify this value to install the components required for VSAM access.

The samples allocation utility produces a summary report that is written to the SYSTSPRT DD specified in the JCL. The report lists the status for allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets and lists the members updated in the *USERHLQ.USERSAMP* data set.

The following figure shows sample output written to SYSTSPRT.

***** Samples Allocation *****
Summary Report

CACCUSX1 compiled on 2012-09-13 15:31:02 by REXX370 3.48
Execution timestamp: 2012-09-13 15:31:02 MVS Product ID: z/OS 01.10.00 SMF ID: ABC System ID: ABC

Effective Parameters:

CACDUNIT: SYSALLDA
CACDVOLM:
CACINHLQ: CEC.V11R1M00
CACMGTCCL:
CACSTGCL:
CACUSHLQ: USER.V11R1M00.CF_IMS
ISPFHLQ: ISP
ISPFLANG: ENU
SERVERROLE: CF_IMS

Data set 'USER.V11R1M00.CF_IMS.USERSAMP' successfully allocated.
Data set 'USER.V11R1M00.CF_IMS.USERCONF' successfully allocated.

Member	Status
-----	-----
CACCUSJ2	Processed successfully in DDN(USERSAMP)

Member	Status
-----	-----
CACCUSP2	Processed successfully in DDN(USERSAMP)

Summary:
Members Successful: 2
Members in Error: 0
Members Not Replaced: 0
Members Processed: 2

Return Status: 0

Figure 1. Sample output for the samples allocation summary report

Installation customization utility

The installation customization utility generates the JCL and configuration members needed in the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets based on the values that you provide in the customization parameters file.

The installation customization utility performs these functions:

- Captures the customization settings that you provide in the customization parameters file *USERHLQ.USERSAMP(CACCUSP2)*.
- Applies the customization parameters to all JCL members associated with the specified *SERVERROLE* parameter and places the customized members in the *USERHLQ.USERSAMP* data set.
- Applies the customization parameters to all configuration members associated with the specified *SERVERROLE* parameter and places the customized members in the *USERHLQ.USERCONF* data set.

You use the *USERHLQ.USERSAMP(CACCUSJ2)* job to run the installation customization utility. You specify the following input as parameters:

CACINHLQ=CAC.V11R1M00

The value specified for the CACINHLQ keyword must be the high-level qualifier for Classic distribution data sets produced by the SMP/E installation. This value is automatically populated with the value previously specified as input to the user samples allocation utility.

CACUSHLQ=USER.V11R1M00

The value specified for the CACUSHLQ keyword must be the high-level qualifier for the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets that were created or updated by the user samples allocation utility. This value is automatically populated with the value previously specified as input to the user samples allocation utility.

MEMBER=(member-name, ...)

This is an optional parameter. The value specified for the MEMBER keyword identifies a list of one or more member names to process. Only a subset of the members associated with the specified SERVERROLE parameter is processed. If you specify multiple member names, you must separate the names with commas and enclose the names in parentheses.

All members are processed when this parameter is not specified.

OVERWRITE=YES | NO

The value specified for the OVERWRITE keyword indicates how to process existing members of target data sets, for example the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets.

- When you specify OVERWRITE=NO, existing members of the target data sets are not replaced. OVERWRITE=NO is the default.
- When you specify OVERWRITE=YES, existing members of the target data sets are replaced.

Example: OVERWRITE=NO is in effect. Members CACCFGDS and CACCFGUT already exist in the target data set. Member CACSX04 does not exist in the target data set.

```
Member  Status
-----  -----
CACCFGDS  CACCFGDS not replaced in DDN(USERSAMP)
CACCFGUT  CACCFGUT not replaced in DDN(USERSAMP)
CACSX04   Processed successfully in DDN(USERSAMP)
```

Example: OVERWRITE=YES is in effect. The processing status for the same members shown in the previous example appear as follows (whether or not any of these members previously existed in the target data set):

```
Member  Status
-----  -----
CACCFGDS  Processed successfully in DDN(USERSAMP)
CACCFGUT  Processed successfully in DDN(USERSAMP)
CACSX04   Processed successfully in DDN(USERSAMP)
```

The processing summary information produced at the bottom of the report identifies the number of members that were stored successfully and the number of members that were not replaced. For example:

```
Summary:
Members Successful:      90
Members in Error:       0
Members Not Replaced:   0
Members Processed:     90
```

SERVERROLE=(*role-name*, ...)

The value of the SERVERROLE keyword specifies that the server environment being installed and customized contains the components required for the Classic data server environment. You can specify one or more roles for your Classic data server. If you specify multiple role names, you must separate the names with commas and enclose the names in parentheses.

CF_ADABAS

Specify this value to install the components required for a Adabas access.

CF_DATACOM

Specify this value to install the components required for CA-Datcom access.

CF_DB2

Specify this value to install the components required for DB2 access.

CF_IDMS

Specify this value to install the components required for CA-IDMS access.

CF_IMS

Specify this value to install the components required for IMS access.

CF_SEQ

Specify this value to install the components required for sequential file access.

CF_VSAM

Specify this value to install the components required for VSAM access.

The utility produces a summary report that is written to the SYSTSPRT DD that is specified in the JCL. The report lists the partitioned data sets and the data set members that were processed. The final summary lists the total number of members processed, the number that were successful, and the number with errors.

The following figure shows sample output written to SYSTSPRT.

***** Installation Customization *****
Summary Report

CACCUSX2 compiled on 2012-08-15 08:46:51 by REXXC370 3.48
Execution timestamp: 2012-08-15 08:49:39 MVS Product ID: z/OS 01.10.00 SMF ID: SYE9 System ID:

Effective Parameters:

CACINHLQ: CEC.V11R1M00
CACUSHLQ: USER.V11R1M00.CF
OVERWRITE: No
SERVERROLE: CF_IMS
CF_VSAM

Processing parameters file: USER.V11R1M00.CF.USERSAMP Member: CACCUSP2

Processing Members for Product: All Role: Common

Member	Status
-----	-----
CACCFGDS	Processed successfully in DDN(USERSAMP)
CACCFGUT	Processed successfully in DDN(USERSAMP)
CACPRTL	Processed successfully in DDN(USERSAMP)
CACLGFLT	Processed successfully in DDN(USERSAMP)
CACXS04	Processed successfully in DDN(USERSAMP)

Processing Members for Product: Classic Federation Role: Common

Member	Status
-----	-----
CACBLDI	Processed successfully in DDN(USERSAMP)
CACCATFG	Processed successfully in DDN(USERSAMP)
CACCATLG	Processed successfully in DDN(USERSAMP)
CACCATMD	Processed successfully in DDN(USERSAMP)
CACCATRP	Processed successfully in DDN(USERSAMP)
. . .	
. . .	

Processing Members for Product: Classic Federation Role: CF_IMS

Member	Status
-----	-----
CACBMP	Processed successfully in DDN(USERSAMP)
CACDBB	Processed successfully in DDN(USERSAMP)
CACSVIMA	Processed successfully in DDN(USERSAMP)
CACSVIMB	Processed successfully in DDN(USERSAMP)
CACSVIMO	Processed successfully in DDN(USERSAMP)
CACIMPAR	Processed successfully in DDN(USERSAMP)
CACIMROT	Processed successfully in DDN(USERSAMP)
. . .	
. . .	

Processing Members for Product: Classic Federation Role: CF_VSAM

Member	Status
-----	-----
CACCDEF	Processed successfully in DDN(USERSAMP)
CACSPCCC	Processed successfully in DDN(USERSAMP)
CACSPCCR	Processed successfully in DDN(USERSAMP)
CACSVVSM	Processed successfully in DDN(USERSAMP)
CACCAPPL	Processed successfully in DDN(USERSAMP)
CACCMODE	Processed successfully in DDN(USERSAMP)
. . .	
. . .	

Summary:

Members Successful: 32
Members in Error: 0
Members Not Replaced: 0
Members Processed: 32

Working with the customization parameters file

These guidelines describe how to enter values in the customization parameters file.

The customization parameters file, *USERHLQ.USERSAMP(CACCUSP2)*, contains pairs of keyword and value settings used to customize JCL and configuration files in the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets.

The following sections provide guidelines for entering input into the customization parameters file and describe how the file is organized. Other considerations include the use of job cards, pre-defined variables, and STEPLIB concatenations. "Customization parameters file settings" on page 13 lists the keyword and value settings that the customization parameters file contains.

Input guidelines

The following guidelines describe how to enter values in the customization parameters file:

- Keyword and value pairs:
 - You cannot change the keyword component.
 - You must delimit the value component with double quotes ("").
 - Spaces are allowed before and after the keyword and value.
 - Values cannot span multiple lines.
- The minimum required parameters that you must change for a successful installation are denoted by an asterisk within parentheses at the end of the comment for that parameter. For example: `CACINHLQ="&CACINHLQ" HLQ of Classic product(*)`
- Comments:
 - An asterisk (*) in column 1 defines the line as a comment line.
 - Any input that you include after the first space after the value component is treated as comments.

File organization

The following table describes the organization of the customization parameters file.

Table 3. Organization of customization parameters file.

Section name	Section content
Common installation	Parameters that apply to all installations, such as the high-level qualifier for the Classic product installation.
Metadata catalog and configuration files	Parameters associated with the metadata catalog and configuration files needed for the data server.
Data server communication parameters	Parameters that define data source name and TCP/IP connection information needed to communicate with a Classic data server environment.
z/OS client parameters	Classic user ID and password used in files that are input into the z/OS sample client.
Security parameters	Security parameters that control user connections to the Classic data server.
Migration and upgrade parameters	Parameters that are specific to the migration of existing installations for each Classic product.

Table 3. Organization of customization parameters file. (continued)

Section name	Section content
IMS	Parameters specific to installations that access IMS data.
CICS® VSAM	Parameters specific to installations that access CICS VSAM data.
DB2	Parameters specific to installations that access DB2 data.
Adabas	Parameters specific to installations that access Adabas data.
CA-IDMS	Parameters specific to installations that access CA-IDMS data.
CA-Datcom	Parameters specific to installations that access CA-Datcom data.
Sequential files	Parameters in the common section apply to sequential files.

Use of job cards

Job card information is defined in the common installation section of the customization parameters file. The following two-line job card information is used as a template when generating JCL members:

```
CACDJOB1="JOB (CLASSIC), 'CLASSIC JOB', CLASS=A, "  
CACDJOB2="MSGCLASS=X, NOTIFY=&SYSUID"
```

The CACDJOB1 value is placed after the job name in each generated JCL member. The CACDJOB2 value is provided on the second line of the job card in each JCL member.

The initial value for the job card keywords is populated from the job card that is specified on the JCL member.

Use of pre-defined variables

Many of the data set values in the customization parameters file contain pre-defined variables such as &CAC to reference previously defined high-level qualifiers. Most of the generated JCL members make use of inline PROC definitions. These variables reference the actual PROC variables. The following table describes what each variable defines:

Table 4. Pre-defined variables.

Variable	Description
&CAC	Classic product installation high-level qualifier
&USERHLQ	User SCACSAMP high-level qualifier
&IMS	IMS installation high-level qualifier

Library concatenations

For Classic data server parameters that require specific DD data set concatenation customization such as STEPLIB, parameters are provided for concatenation. You can specify the same parameter keyword multiple times. The order specified for the parameter keywords is the order in which the data sets will be included in the data set concatenation.

Customization parameters file settings:

The parameter keyword and value pairs in the customization parameters file are set to default values. You can modify these values to customize your installation.

The following table lists the parameters in each section of the customization parameters file, the parameter default values, and a description of each parameter.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2).

Parameter	Default value	Description
Common section		
CACINHLQ	CAC.V11R1M00	High-level qualifier of the installation data sets for the Classic product. This value is populated with the value specified for the CACINHLQ input parameter of the CACCUSJ1 job.
CACUSHLQ	USER.V11R1M00	High-level qualifier for the <i>USERHLQ.USERSAMP</i> and <i>USERHLQ.USERCONF</i> data sets. This value is populated with the value specified for the CACUSHLQ input parameter of the CACCUSJ1 job. This value is also the default high-level qualifier for the metadata catalog and configuration files for the Classic data server.
CACDUNIT	SYSALLDA	Disk unit that is used for the generated jobs that create data sets such as the configuration files for the Classic data server. This value is populated with the value specified for the CACDUNIT input parameter of the CACCUSJ1 job. If the value is " ", it is assumed that the site SMS rules will determine the data set allocation.
CACDVOLM	" "	Disk volume that is used for the generated jobs that create data sets such as the configuration files for the Classic data server. This value is populated with the value specified for the CACDVOLM input parameter of the CACCUSJ1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.
CACSTGCL	" "	SMS storage class that is used for the generated jobs that create data sets such as the configuration files for the Classic data server. This value is populated with the value specified for the CACSTGCL input parameter of the CACCUSJ1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.
CACMGTCL	" "	SMS management class that is used for the generated jobs that create data sets such as the configuration files for the Classic data server. This value is populated with the value specified for the CACMGTCL input parameter of the CACCUSJ1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
CACDJOB1	" "	Pre-populated value from the first line of the JOBCARD provided on the CACCUSJ1 job. The value is substituted into all generated JCL.
CACDJOB2	" "	Pre-populated value from the second line of the JOBCARD provided on the CACCUSJ1 job. The value is substituted into all generated JCL.
Metadata catalog and configuration files		
CACADMUS	CACUSER	User ID to which to grant SYSADM privileges are granted in the metadata catalog.
CATLGMB	150	Megabytes for the primary catalog allocation. For guidance in determining the actual number of MB you require, see Estimating the size of the metadata catalog.
CATPATH	/opt/IBM/isclassic111/catalog	USS file system path to the metadata catalog files. The named directory contains the following file names: cacat Data component cacindx Index component zFS file system resident catalogs are recommended (rather than other file systems, such as physical sequential). When this parameter is specified, it supersedes CACCATNM and CACIDXNM and NEWCATNM and NEWIDXNM.
CACCATNM	&USRHLQ..CATALOG	Suffix for the Version 11.1 metadata catalog data file created during the installation customization process. The &USRHLQ value is replaced in the generated JCL PROC with the high-level qualifier specified for the CACUSHLQ parameter.
CACIDXNM	&USRHLQ..CATINDEX	Suffix for the Version 11.1 metadata catalog index file created during the installation customization process. The &USRHLQ value is replaced in the generated JCL PROC with the high-level qualifier specified for the CACUSHLQ parameter.
CACCFGNM	&USRHLQ..CACCFGD	Suffix for the Version 11.1 configuration data file created during the installation customization process. The &USRHLQ value is replaced in the generated JCL procedure with the high-level qualifier specified for the CACUSHLQ parameter.
CACCFGIX	&USRHLQ..CACCFGX	Suffix for the Version 11.1 configuration index file created during the installation customization process. The &USRHLQ value is replaced in the generated JCL procedure with the high-level qualifier specified for the CACUSHLQ parameter.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
DIAGLGST	CF.DIAGLOG	z/OS log stream name for the diagnostic log for the Classic data server. If a log stream name is not specified, the log service is configured to write to the CACLOG DD data set which is a temporary data set.
DIAGLGDS	Y	Identifies whether the z/OS log stream should use DASD or the coupling facility: <ul style="list-style-type: none"> • Y: DASD • N: Coupling facility This value is valid when DIAGLGST is specified.
DIAGLGRT	7	Retention period, in days, to retain the log records before they are eligible to be deleted. This value is valid when DIAGLGST is specified.
DIAGLGSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when DIAGLGST is specified.
DIAGLGSR	CCL1	Coupling facility structure name (STRUCTNAME). This value is valid when DIAGLGST is specified and the coupling facility is chosen (DIAGLGDS="N").
Classic data server communication parameters		
DSDSRCE	CACSAMP	Data source name for the query processor service. This is the data source name that clients use to connect to the Classic data server.
DSHOST	0.0.0.0	Host name or IP address where the Classic data server will run.
DSPORT	9087	Port number on which the connection handler service for the Classic data server will listen. This listen port is used to communicate with the CDA and other client applications.
z/OS client parameters		
CLNTUSER	CACUSER	User ID to use in z/OS client application samples such as the metadata utility connection string and the sample SQL file.
CLNTPSWD	CACPWD	Password associated with the user ID provided for the CLNTUSER parameter.
Security parameters		
CFSAFX	CACSX04	SAFEXIT load module that enables security for the Classic data server. Specifying a value enables security for the Classic data server. A value of "" disables security for the Classic data server.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
CFSAFVLD	N	VALIDATE=Y/N parameter on the query processor service for the SAFEXIT, when the exit is enabled by using the CFSAFX customization parameter. This value instructs the service to validate the access authority of a user by using additional security parameters: CFSAFNET - NETACCESS CFSAFSPC - Stored procedure SPCLASS CFSAFIMC - IMS PSB schedule class CFSAFIMP - IMS PSB prefix CFSAFDCE - CA-Datacom EXCLUDE CFSAFADC - Adabas ADACLASS
CFSAFNET	None	NETACCESS parameter on the query processor service for the SAFEXIT, when the exit is enabled and the CFSAFVLD customization parameter = Y. This value (Y/N) specifies if IP address validation is enabled.
CFSAFSPC	None	SPCLASS parameter on the query processor service for the SAFEXIT, when the exit is enabled and the CFSAFVLD customization parameter = Y. This value specifies the name of a class that is used to check for RACF-authorized use of stored procedure names.

Migration and upgrade parameters : The parameters in this section apply to installations that you need to migrate from a previous release of Classic federation to Version 11.1. If a migration is not required, use the default values. When migrating or upgrading, new catalog information is obtained from the parameters defined in the 'Metadata catalog and configuration files' section above.

OLDCAHLQ	CAC.V10R1M00	High-level qualifier for the pre-version 11.1 product installation.
OLDCTHLQ	CAC.V10R1M00	High-level qualifier for the pre-version 11.1 metadata catalog that is being upgraded.
OLDCPATH	None	If upgrading an existing zFS 11.1 catalog, use this parameter. When defined, this parameter supersedes OLDCATNM and OLDIDXNM. Specifies the USS file system path to the metadata catalog files. (for example, /opt/IBM/isclassic111/catalog). The named directory contains the following file names: cacat Data component cacindx Index component
OLDCATNM	&OLDCAT..CATALOG	Suffix for the pre-version 11.1 metadata catalog data file. The &OLDCAT value will be replaced in the generated JCL PROC with the high-level qualifier specified for the OLDCTHLQ parameter.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
OLDCATIX	&OLDCAT..CATINDEX	Suffix for the pre-version 11.1 metadata catalog index file. The &OLDCAT value will be replaced in the generated JCL PROC with the high-level qualifier specified for the OLDCTHLQ parameter.
OLDCFHLQ	CAC.V10R1M00	High-level qualifier for the pre-version 11.1 configuration files that are being upgraded. <ul style="list-style-type: none"> • If this value specifies a release prior to version 9.5 it will be the HLQ for the SCACCONF data set. • If this value specifies a version 9.5 release it will be the HLQ for the Classic data server configuration files (CACCFGD and CACCFGX).
OLDCFGNM	&OLDCFG..SCACCONF	If you are migrating from a release prior to version 9.5, this value specifies the SCACCONF data set that contains the configuration member. This data set is associated with the VHSCONF DD statement in the pre-version 9.5 startup JCL for the Classic data server.
OLDDSCFG	CACDSCF	If you are migrating from a release prior to version 9.5, this value specifies the configuration member name in the SCACCONF data set. This member is associated with the VHSCONF DD statement in the pre-version 9.5 startup JCL for the Classic data server.
OLDCFGNM	&OLDCFG..CACCFGNM	Suffix for the post-version 9.1 configuration data file. The & OLDCFG value is replaced in the generated JCL PROC with the high-level qualifier specified for the OLDCFHLQ parameter.
OLDCFGIX	&OLDCFG..CACCFGIX	Suffix for the post-version 9.1 configuration index file. The & OLDCFG value is replaced in the generated JCL PROC with the high-level qualifier specified for the OLDCFHLQ parameter.
NEWCFGNM	&NEWCFG..CACCFGNM	Suffix for the new version 11.1 configuration data file. The & NEWCFG value is replaced in the generated JCL PROC with the high-level qualifier specified for the NEWCTHLQ parameter.
NEWCFGIX	&NEWCFG..CACCFGIX	Suffix for the new version 11.1 configuration index file. The & NEWCFG value is replaced in the generated JCL PROC with the high-level qualifier specified for the NEWCTHLQ parameter.

IMS parameters: The parameters in this section are specific to Classic data servers that access IMS. See Setting up access to IMS for details about configuring Classic data servers for IMS access.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
IMSINHLQ	IMS	High-level qualifier (HLQ) for THE IMS installation libraries. This HLQ is applied to the IMS data sets in the STEPLIB sections of the generated JCL. It replaces the references to &IMS. below in other keyword values such as IMSSTEPL.
IMSDFPSB	DEFPSB	Default PSB that the DRA initialization service or the ODBA service uses to access IMS.
IMSDRAUS	DRAUSER	User ID that the DRA initialization service uses to access IMS.
IMSSSID	SSID	IMS subsystem ID that the ODBA initialization service uses
IMSPZPSF	00	DRA startup table suffix (DFSPZPxx) that the DRA initialization service uses.
IMSSTEPL	&IMS..SDFSRESL	STEPLIB concatenation for files needed to access IMS. The IMSTEPL is a keyword that you can specify multiple times. It provides a STEPLIB concatenation. The order of the multiple IMSSTEPL keywords defines the order in which the files are included in the STEPLIB concatenation for the generated JCL members. The &IMS value is replaced in the generated JCL PROC with the high-level qualifier specified for the IMSINHLQ parameter.
IMSDBDLB	&CICS..DBDLIB	DBDLIB used by the Classic data server to locate DBDs during CREATE TABLE processing. The IMSDBDLB is a keyword that you can specify multiple times. It provides a DBDLIB concatenation. The order of the multiple IMSDBDLB keywords defines the order in which the files is included in the DBDLIB concatenation for generated the generated JCL members. The &IMS value is replaced in the generated JCL PROC with the high-level qualifier specified for the IMSINHLQ parameter.
CFSAFIMC	" "	IMS PSB schedule class parameter on the query processor service for the SAFEXIT when the exit is enabled and the CFSAFVLD customization parameter = Y. This parameter specifies the name of the RACF® resource class that is checked to determine whether the user has authority to schedule or access the PSBs associated with the tables referenced in a query

CICS VSAM parameters The parameters in this section are specific to Classic data servers that access CICS VSAM files. See Setting up access to CICS VSAM for details about configuring Classic data servers for CICS VSAM access.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
VSCINHLQ	CICSTSn.CICS	High-level qualifier (HLQ) for the CICS libraries. This HLQ is applied to the CICS data sets in the STEPLIB sections of the generated JCL. It replaces the references to &CICS.. below in other keyword values such as VSCSTEPL and VSCDFHCS.
VSCSTEPL	&CICS..SDFHLOAD	STEPLIB concatenation for files needed to access CICS. The VSCSTEPL is a keyword that you specify multiple times. It provides a STEPLIB concatenation. The order of the multiple VSCSTEPL keywords defines the order in which the files are included in the STEPLIB concatenation for the generated JCL members. The &CICS value is replaced in the generated JCL PROC with the high-level qualifier specified for the VSCINHLQ parameter.
VSCDFHCS	&CICS.. DBDLIB	DFHCS data set to use when defining the Classic federation programs and transactions to CICS.
VSCNETNM	CACCICS1	CICS NETNAME to use when defining the Classic federation programs and transactions to CICS.

DB2 parameters: The parameters in this section are specific to Classic data servers that access DB2. See Setting up access to DB2 for z/OS for details about configuring Classic data servers for DB2 access.

DB2INHLQ	DB2	High-level (HLQ) qualifier for the DB2 libraries. This HLQ is applied to the DB2 data sets in the STEPLIB sections of the generated JCL. It replaces the references to &DB2.. below in other keyword values such as DB2STEPL.
DB2PLAN	CAC10PLN	DB2 plan name that the DB2 call attachment facility (CAF) access service uses.
DB2DSN	DSN	DB2 subsystem ID that the DB2 CAF access service uses.
DB2STEPL	&DB2..SDSNLOAD	DB2 STEPLIB concatenation for files needed to access DB2. The DB2STEPL is a keyword that you can specify multiple times. It provides a STEPLIB concatenation. The order of the multiple DB2STEPL keywords defines the order in which the files are included in the STEPLIB concatenation for the generated JCL members. The &DB2 value is replaced in the generated JCL PROC with the high-level qualifier specified for the DB2INHLQ parameter.

Adabas parameters The parameters in this section are specific to Classic data servers that access Adabas. See Setting up access to Adabas databases for details about configuring Classic data servers for Adabas access.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
ADAINHLQ	ADABAS	High-level (HLQ) qualifier for the Adabas libraries. This HLQ is applied to the Adabas data sets in the STEPLIB sections of the generated JCL. It replaces the references to &ADA.. below in other keyword values such as ADASTEPL.
ADAEPSVC	251	SVC number of the Adabas nucleus.
ADAEPDID	32	Default database ID for Adabas.
ADALNK8	Y	Version of the Adabas system that is being accessed. The values are either version 8 or a prior version. <ul style="list-style-type: none"> • Y: Indicates version 8 or higher. • N: Indicates a version prior to version 8.
ADASTEPL	&ADA..LOAD	STEPLIB concatenation for files needed to access Adabas. The ADASTEPL is a keyword that you can specify multiple times. It provides a STEPLIB concatenation. The order of the multiple ADASTEPL keywords defines the order in which the files are included in the STEPLIB concatenation for the generated JCL members. The &ADA value is replaced in the generated JCL PROC with the high-level qualifier specified for the ADAINHLQ parameter.
CFSAFADC	None	ADAClass parameter on the query processor service for the SAFEXIT when the exit is enabled and the CFSAFVLD customization parameter = Y.

CA-IDMS parameters: The parameters in this section are specific to Classic data servers that access CA-IDMS. See Setting up access to CA-IDMS for details about configuring Classic data servers for CA-IDMS access.

IDMINHLQ	IDMS	High-level qualifier (HLQ) for the CA-IDMS libraries. This HLQ is applied to the CA-IDMS data sets in the STEPLIB sections of the generated JCL. It replaces the references to &IDMS.. below in other keyword values such as IDMSYCTL and IDMSTEPL.
IDMSYCTL	&IDMS..SYSCTL	CA-IDMS SYSCTL file that the Classic data server uses to access the CA-IDMS central version. The &IDMS value is replaced in the generated JCL PROC with the high-level qualifier specified for the IDMINHLQ parameter.
IDMGDMCL	GLBLDMCL	Global DMCL that the CA-IDMS central version uses. This value is specified in the SYSIDMS setting for the Classic data server. This parameter allows the CA-IDMS client layer to obtain access to the DDLMSG area to report errors if necessary.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
IDMDCMSG	&IDMS..SYSMSG.DDLDCMSG	Local data set for the DDLDCMSG area that the CA-IDMS central version uses. This parameter allows the CA-IDMS client layer to obtain access to the DDLMSG area to report errors if necessary. The &IDMS value is replaced in the generated JCL PROC with the high-level qualifier specified for the IDMINHLQ parameter.
IDMACLIB	&IDMS..DISTMAC	CA-IDMS MACLIB. This parameter is used when setting up security for accessing CA-IDMS from Classic federation to relink a new IDMSSTRT module. The &IDMS value is replaced in the generated JCL PROC with the high-level qualifier specified for the IDMINHLQ parameter.
IDMSTEPL	&IDMS..DBA.LOADLIB	STEPLIB concatenation for files needed to access CA-IDMS. The IDMSTEPL is a keyword that you can specify multiple times. It provides a STEPLIB concatenation. The order of the multiple IDMSTEPL keywords defines the order in which the files are included in the STEPLIB concatenation for the generated JCL members. The &IDMS value is replaced in the generated JCL PROC with the high-level qualifier specified for the IDMINHLQ parameter.
CA-Datacom parameters: The parameters in this section are specific to Classic data servers that access CA-Datacom. See Setting up access to CA-Datacom for details about configuring Classic data servers for CA-Datacom access.		
DCMINHLQ	DCOM	High-level qualifier (HLQ) for the CA-Datacom libraries. This HLQ is applied to the CA-Datacom data sets in the STEPLIB sections of the generated JCL. It replaces the references to &DC.. below in other keyword values such as DCMACLIB and DCMSTEPL.
DCMACLIB	&DC..CAI.DATAACOM.THLQ.CAIMAC	CA-Datacom MACLIB. This parameter is used to assemble the sample CACDCURT User Requirements Table. The &DC value is replaced in the generated JCL PROC with the high-level qualifier specified for the DCMINHLQ parameter.
DCMDDIDT	&USRHLQ..SCACSAMP(CACDCID)	DDIDENT data set that the Classic data server uses. It contains a user ID and password to connect to the CA-Datacom dictionary search facility. The Classic data server accesses the CA-Datacom dictionary during CREATE TABLE processing.
DCMUSER	DATAACOM-INSTAL	CA-Datacom user ID to use when accessing the CA-Datacom dictionary search facility. This user ID is placed in the DDIDENT data set specified by the DCMDDIDT parameter.

Table 5. Parameter and default settings for SCACSAMP (CACCUSP2). (continued)

Parameter	Default value	Description
DCMPASWD	NEWUSER	Password for the CA-Datcom user ID to use when accessing the CA-Datcom dictionary search facility. This password is placed in the DDIDENT data set specified by the DCMDDIDT parameter.
DCMSTEPL	&DC..CAI.CAILIB &DC..DATACOM.CHLQ.CUSLIB &DC..DATACOM.THLQ.CAILIB	STEPLIB concatenation for files needed to access CA-Datcom. The DCMSTEPL is a keyword that you can specify multiple times. It provides a STEPLIB concatenation. The order of the multiple DCMSTEPL keywords defines the order in which the files are included in the STEPLIB concatenation for the generated JCL members. The &DC value is replaced in the generated JCL PROC with the high-level qualifier specified for the DCMINHLQ parameter.
CFSAFDCE	None	EXCLUDE parameter on the query processor service for the SAFEXIT when the exit is enabled and the CFSAFVLD customization parameter = Y. This parameter specifies that the query processor service should not provide an ACEE address in commands sent to CA-Datcom.
Language Environment® (LE) and compiler parameters		
CACLEHLQ	SYS1	High-level qualifier for the LE runtime SCEELKED data set.
CACCBCMP	IGYCRCTL	COBOL compiler name to use when compiling a stored procedure.
CACCBHLQ	SYSL	High-level qualifier for the SIGYCOMP data set.
CACCSTEP	&COBOL..SIGYCOMP	COBOL libraries to include in the STEPLIB for compile sample members. CACCSTEP is a keyword that you can specify multiple times. This keyword provides a STEPLIB concatenation. The order of the multiple CACCSTEP keywords defines the order in which the files is included in the STEPLIB concatenation for the generated JCL members. The &COBOL value is replaced in the generated JCL PROC with the high-level qualifier specified for the CACCBHLQ parameter.

Installing Classic data servers

The installation customization steps that you follow depend on the product that you are using and if you are creating a new installation or upgrading to Version 11.1 from a previous version of Classic federation.

About this task

Procedure

Follow the appropriate installation customization steps for either a new installation or an installation upgrade from an earlier version of Classic federation:

Procedure

- If you are creating a new Version 11.1 installation, follow the installation customization steps for a new Classic federation installation.
- If you are upgrading to Version 11.1 from a previous version, follow the steps for upgrading Classic federation.

Installing a new Classic federation data server

You can follow the installation customization process to install and customize a Classic federation data server.

Before you begin

Before you begin the installation customization process, you must complete the SMP/E installation and the steps required to “Setting up the installation environment” on page 2.

About this task

When setting up the metadata catalog, the default configuration creates and initializes the metadata catalog as a zSeries File System (zFS) file. The z/FS file provides significant performance and capacity improvements compared to the sequential or linear data set formats used in releases prior to V11.1.

The topic *Creating and initializing zFS metadata catalogs* provides more information about the use of zFS metadata catalogs.

Procedure

1. Edit the user samples allocation utility JCL in the installation samples member SCACSAMP(CACCUSJ1). Follow the instructions in the JCL to edit the job card and procedure variables. Then specify input parameters to the samples allocation utility. You specify the following input parameters:

CACINHLQ=CAC.V11R1M00

The value specified for the CACINHLQ keyword must be the high-level qualifier of the installation data sets that the SMP/E installation produces for Classic federation.

CACUSHLQ=USER.V11R1M00

The value specified for the CACUSHLQ keyword is the high-level qualifier for the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets that the samples allocation utility creates or updates.

CACDUNIT=SYSALLDA

The value specified for the CACDUNIT keyword identifies the disk unit that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter. You do not need to specify a value for CACDUNIT if SMS manages the data sets.

CACDVOLM=

The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ.USERSAMP* and

USERHLQ.USERCONF data sets. This is an optional parameter. You do not need to specify a value for *CACDVOLM* if SMS manages the data sets.

CACSTGCL=

The value specified for the *CACSTGCL* keyword identifies the SMS storage class that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter. Specify a value for *CACSTGCL* only when a specific storage class is required.

CACMGTCCL=

The value specified for the *CACMGTCCL* keyword identifies the SMS management class that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter. Specify a value for *CACMGTCCL* only when a specific management class is required.

ISPFHLQ=ISP

The value specified for the *ISPFHLQ* keyword identifies the high-level qualifier for ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU

The value specified for the *ISPFLANG* keyword identifies the language prefix for the ISPF installation.

SERVERROLE=role-name

The value specified for the *SERVERROLE* keyword identifies the server environment to install and customize. The role determines the JCL members that are customized and the services that are configured for the Classic data server. The list below contains the possible values. You can specify multiple values at the same time by using the following syntax:

SERVERROLE=(role-name1, role-name2)

Valid values for role-name:

SERVERROLE value	Installs components for ...
CF_ADABAS	Adabas access
CF_DATACOM	CA-Datcom
CF_DB2	DB2 access
CF_IDMS	CA-IDMS access
CF_IMS	IMS access
CF_SEQ	Sequential file access
CF_VSAM	VSAM access

2. Submit *SCACSAMP(CACCUSJ1)* to allocate the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. Verify that all job steps result in a return code <= 4. This job populates the *USERHLQ.USERSAMP* data set with the necessary objects and the customization parameters file for Classic federation, *CACCUSP2* .
3. Edit the Classic federation customization parameters file *USERHLQ.USERSAMP(CACCUSP2)* to provide customization parameters.

This file will contain only the parameters that apply to the specified SERVERROLE. See the customization parameters file settings for details.

4. Submit the generated *USERHLQ.USERSAMP(CACCUSJ2)* customization utility JCL. Verify that all job steps result in a return code ≤ 4 . The *USERHLQ.USERSAMP* data set is populated with the customized JCL and objects needed to run a Classic data server for the specified SERVERROLE.
5. Optional: Define the z/OS log stream for the diagnostic log for the Classic federation data server. This step is only needed when the *DIAGLGST* or *EVENLGST* parameter in the *CACCUSP2* files specifies a log stream name.
 - a. Verify that you have the authority required to run the Administrative Data Utility (*IXCMIAPU*). The job that defines the logs runs this utility.
 - b. Submit the generated *USERHLQ.USERSAMP(CACCFSL)* JCL to define the log stream and logs.
 - c. Verify that all job steps result in a return code = 0.
6. Optional: If the *CATPATH* value specified that the metadata catalog files will reside in a new zFS file system aggregate, you must define and format the new zFS aggregate.
 - a. Verify that you have the authority required to run the following jobs:
 - SAF READ-access level authorization is required for the *SUPERUSER.FILESYS.PFSC* resource in the *UNIXPRIV* class to run the zFS administration command, *IOEZADM*.
 - SAF READ-access level authorization is required for the *SUPERUSER.FILESYS.MOUNT* resource in the *UNIXPRIV* class to perform *MOUNT* and *UNMOUNT* operations against USS file systems.
 - b. View and edit *USERHLQ.USERSAMP(CECCRZCT)* job cards.
 - c. Submit *USERHLQ.USERSAMP(CECCRZCT)*.
 - d. Verify that all job steps result in a return code = 0.
 - e. Mount the file systems using the sample *MOUNT* command provided in *USERHLQ.USERSAMP(CECCRZCT)*.
7. Submit the generated *USERHLQ.USERSAMP(CACCATFG)* JCL to allocate and initialize the metadata catalog and allocate the configuration files for the Classic data server. This job populates the configuration with the service definitions required for the specified SERVERROLE.
8. Submit the generated *USERHLQ.USERSAMP(CACDS)* JCL to start the Classic data server.
9. Run the validation job in the next step if security is enabled for the Classic federation data server. If security is not enabled, skip this step and continue with the next step to run the validation job.

Security is enabled for the Classic data server by setting the *CFSAFX* parameter in the installation customization parameters file. If security is enabled, the following steps are required before you run the validation job in the final step. If security is not enabled, continue with the next step.

Enabling security requires providing a password for the user access. The utilities used in the validation job require encrypted passwords to access the server. Set up the encrypted password for the validation job by following these steps:

- a. Edit the generated *USERHLQ.USERCONF(CACPWDIN)* member. This member provides a *PASSWORD=value* parameter for the password generator utility. Set the value to the TSO password for the User ID that you use to run the customization jobs.

- b. Submit the generated *USERHLQ.USERSAMP(CACENCRP)* JCL to run the password generator utility. This JCL updates the *USERHLQ.USERCONF(CACPWDIN)* with the encrypted value of the password provided in the previous step.
 - c. Edit the *USERHLQ.USERCONF(CACPWDIN)* member and copy the hex string value (x'<16-byte hexadecimal value>') for the ENCRYPTED= keyword.
 - d. Edit the generated *USERHLQ.USERCONF(CACMUCON)* member and replace the X'.ENCRYP PASSWD..' string with the hex string copied in the previous step.
 - e. Edit the generated *USERHLQ.USERSAMP(CACQRSYS)* member and replace the second line of the member with the hex string that you copied to ensure that the hex string starts in the first column
10. Submit the generated *USERHLQ.SCACSAMP(CACCUSVF)* validation job. Verify that all job steps result in a return code <= 4.
 11. You can now configure the Classic Data Architect to access the Classic data server, select from the sample tables, and map additional tables.

Upgrading a Classic federation installation

To upgrade a Classic federation environment to Version 11.1 from a previous version, complete the steps in the update process.

Procedure

1. Edit the user samples allocation utility JCL in the installation samples member *SCACSAMP(CACCUSJ1)*. Follow the instructions in the JCL to edit the job card and procedure variables. Then specify input parameters to the samples allocation utility. You specify the following input parameters:

CACINHLQ=CAC.V11R1M00

The value specified for the CACINHLQ keyword must be the high-level qualifier of the installation data sets that the SMP/E installation produces for Classic federation.

CACUSHLQ=USER.V11R1M00

The value specified for the CACUSHLQ keyword is the high-level qualifier for the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets that the samples allocation utility creates or updates.

CACDUNIT=SYSALLDA

The value specified for the CACDUNIT keyword identifies the disk unit that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter. You do not need to specify a value for CACDUNIT if SMS manages the data sets.

CACDVOLM=

The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter. You do not need to specify a value for CACDVOLM if SMS manages the data sets.

CACSTGCL=

The value specified for the CACSTGCL keyword identifies the SMS storage class that is used when allocating the *USERHLQ.USERSAMP*

and *USERHLQ.USERCONF* data sets. This is an optional parameter. Specify a value for CACSTGCL only when a specific storage class is required.

CACMGTCCL=

The value specified for the CACMGTCCL keyword identifies the SMS management class that is used when allocating the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. This is an optional parameter. Specify a value for CACMGTCCL only when a specific management class is required.

ISPFHLQ=ISP

The value specified for the ISPFHLQ keyword identifies the high-level qualifier for ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU

The value specified for the ISPFLANG keyword identifies the language prefix for the ISPF installation.

SERVERROLE=role-name

The value specified for the SERVERROLE keyword identifies the server environment to install and customize. The role determines the JCL members that are customized and the services that are configured for the Classic data server. The list below contains the possible values. You can specify multiple values at the same time by using the following syntax:

`SERVERROLE=(role-name1, role-name2)`

Valid values for role-name:

SERVERROLE value	Installs components for ...
CF_ADABAS	Adabas access
CF_DATACOM	CA-Datcom
CF_DB2	DB2 access
CF_IDMS	CA-IDMS access
CF_IMS	IMS access
CF_SEQ	Sequential file access
CF_VSAM	VSAM access

2. Submit SCACSAMP(CACCUSJ1) to allocate the *USERHLQ.USERSAMP* and *USERHLQ.USERCONF* data sets. Verify that all job steps result in a return code <= 4. This job populates the *USERHLQ.USERSAMP* data set with the necessary objects and the customization parameters file for Classic federation, CACCUSP2 .
3. Edit the Classic federation customization parameters file *USERHLQ.USERSAMP(CACCUSP2)* to provide customization parameters. This file will contain only the parameters that apply to the specified SERVERROLE. See the customization parameters file settings for details.
4. Submit the generated *USERHLQ.USERSAMP(CACCUSJ2)* customization utility JCL. Verify that all job steps result in a return code <= 4. The *USERHLQ.USERSAMP* data set is populated with the customized JCL and objects needed to run a Classic data server for the specified SERVERROLE.

5. Optional: Define the z/OS log stream for the diagnostic log for the Classic federation data server. This step is only needed when the DIAGLGST or EVENLGST parameter in the CACCUSP2 files specifies a log stream name
 - a. Verify that you have the authority required to run the Administrative Data Utility (IXCMIAPU). The job that defines the logs runs this utility.
 - b. Submit the generated *USERHLQ.USERSAMP(CACCFSL)* JCL to define the log stream and logs.
 - c. Verify that all job steps result in a return code = 0.
6. Optional: If the system catalog files reside in a new zFS file system aggregate, you must define and format the new zFS aggregate.
 - a. Verify that you have the authority required to run the following jobs:
 - SAF READ-access level authorization is required for the SUPERUSER.FILESYS.PFSCCTL resource in the UNIXPRIV class to run the zFS administration command, IOEZADM.
 - SAF READ-access level authorization is required for the SUPERUSER.FILESYS.MOUNT resource in the UNIXPRIV class to perform MOUNT and UNMOUNT operations against USS file systems.
 - b. View and edit *USERHLQ.USERSAMP(CECCRZCT)* job cards.
 - c. Submit *USERHLQ.USERSAMP(CECCRZCT)*.
 - d. Verify that all job steps result in a return code = 0.
7. Submit the generated *USERHLQ.USERSAMP(CACCATUP)* member to upgrade the existing metadata catalog.
8. Allocate the configuration files and migrate the configuration.
 - If you are migrating from a Version 10.1 release or later, submit the generated *USERHLQ.USERSAMP(CACCFGM1)* member.
 - If you are migrating from a Version 9.5 release, submit the generated *USERHLQ.USERSAMP(CACCFGMD)* member.
 - If you are migrating from a pre-Version 9.5 release, submit the generated *USERHLQ.USERSAMP(CACCFGMV)* member.

These jobs allocate the configuration files for the Classic data server and run the configuration migration and maintenance utility, *CACCFGUT*, to migrate the configuration.

9. Submit the generated *USERHLQ.USERSAMP(CACDS)* JCL to start the Classic data server.
10. Run the validation job in the next step if security is enabled for the Classic federation data server. If security is not enabled, skip this step and continue with the next step to run the validation job.

Security is enabled for the Classic data server by setting the *CFSAFX* parameter in the installation customization parameters file. If security is enabled, the following steps are required before you run the validation job in the final step. If security is not enabled, continue with the next step.

Enabling security requires providing a password for the user access. The utilities used in the validation job require encrypted passwords to access the server. Set up the encrypted password for the validation job by following these steps:

- a. Edit the generated *USERHLQ.USERCONF(CACPWDIN)* member. This member provides a *PASSWORD=value* parameter for the password generator utility. Set the value to the TSO password for the User ID that you use to run the customization jobs.

- b. Submit the generated *USERHLQ.USERSAMP(CACENCRP)* JCL to run the password generator utility. This JCL updates the *USERHLQ.USERCONF(CACPWDIN)* with the encrypted value of the password provided in the previous step.
 - c. Edit the *USERHLQ.USERCONF(CACPWDIN)* member and copy the hex string value (x'<16-byte hexadecimal value>') for the ENCRYPTED= keyword.
 - d. Edit the generated *USERHLQ.USERCONF(CACMUCON)* member and replace the X'.ENCRYP PASSWD..' string with the hex string copied in the previous step.
 - e. Edit the generated *USERHLQ.USERSAMP(CACQRSYS)* member and replace the second line of the member with the hex string that you copied to ensure that the hex string starts in the first column
11. Submit the generated *USERHLQ.SCACSAMP(CACCUSVF)* validation job. Verify that all job steps result in a return code <= 4.
 12. You can now configure the Classic Data Architect to access the Classic data server, select from the sample tables, and map additional tables.

Installing the Classic Data Architect

To install the Classic Data Architect you extract an installation zip package and run IBM Installation Manager.

Before you begin

If you have installed an earlier beta version of the Classic Data Architect (CDA) Version 11.1, you must uninstall it first. See “Uninstalling the Classic Data Architect” on page 31.

Ensure that your client computer meets the following minimum system requirements:

Operating system:

- Microsoft Windows 7
- Microsoft Windows Vista

Memory

1024 MB.

Disk space

800 MB for both IBM Installation Manager and the Classic Data Architect.

Procedure

Start the CDA installation as a non-Administrator or as an Administrator.

Method	Description
To start the CDA installation as a non-Administrator	<ol style="list-style-type: none">1. Unzip the installation package to a temporary directory.2. From the command line, change to the temporary directory and take one of the following actions:<ul style="list-style-type: none">• On Windows: Run <code>userinst.exe</code>.• On Linux: Run <code>userinst</code>.3. Proceed through the Installation Manager wizard.
To start the CDA installation as an Administrator	<ol style="list-style-type: none">1. Unzip the installation package to a temporary directory.2. From the command line, change to the temporary directory and take one of the following actions:<ul style="list-style-type: none">• On Windows: Run <code>install.exe</code>.• On Linux: Run <code>install</code>.3. Proceed through the Installation Manager wizard.

Results

When the `userinst` or `install` program starts, Installation Manager is installed if it is not already on your computer. Installation Manager is configured with the location of the repository (installation files) for IBM InfoSphere Classic Data Architect V11.1.

What to do next

You can launch the product:

- From Windows: **Start > Programs > IBM Classic Data Architect > IBM InfoSphere Classic Data Architect V11.1**. Alternatively, from a command line run `<installRoot>/eclipse.exe`.
- From Linux: **Applications > IBM Classic Data Architect > IBM InfoSphere Classic Data Architect V11.1**. Alternatively, from a command line run `<installRoot>/eclipse`.

Starting IBM Installation Manager

If you start the Classic Data Architect installation from the downloadable image, IBM Installation Manager starts automatically.

About this task

When you start the installation of Classic Data Architect from the downloadable image, either by running the `install` or `userinst` program, IBM Installation Manager automatically starts even if it is not already installed.

If you already installed Installation Manager, you can start it by using one of the methods in the following procedure.

Procedure

Start the Installation Manager from Windows or Linux.

Method	Description
To start the Installation Manager from Windows	Click Start > All Programs > IBM Installation Manager > IBM Installation Manager .
To start the Installation Manager from Linux	Click Applications > IBM Installation Manager > IBM Installation Manager or alternatively change to <i>Installation Manager directory /eclipse</i> and run <i>IBMIM</i> .

Uninstalling the Classic Data Architect

You can uninstall the Classic Data Architect by using the IBM Installation Manager.

Before you begin

To uninstall the IBM InfoSphere Classic Data Architect product package, you must log in to the system by using the same user account that you used to install the product package. You must close the programs that you installed by using IBM Installation Manager.

About this task

You can use the **Uninstall** option in IBM Installation Manager to uninstall the IBM InfoSphere Classic Data Architect product package from a single installation location. You can also uninstall all of the installed packages from every installation location.

To uninstall the CDA product package, complete the steps in the following procedure.

Procedure

1. Start IBM Installation Manager.
2. On the Start page, click the **Uninstall** button.
3. On the Uninstall Packages page, from the **Installation Packages** list, select IBM InfoSphere Classic Data Architect version 11.1.0, and click **Next**.
4. On the Summary page, review the list of packages that will be uninstalled. The Complete page displays after the packages are removed.
5. Click **Finish**.

Installing the ODBC/CLI and JDBC clients

Install the ODBC driver and CLI, the JDBC client, or both on Linux, UNIX, and Windows systems to connect workstation client applications to the IBM InfoSphere Classic products on the mainframe.

Before you begin

Uninstall all beta versions of Classic ODBC/CLI and JDBC.

About this task

The ODBC/CLI and JDBC clients support both 32-bit and 64-bit operating systems on most platforms. Exceptions are noted on the following list:

- AIX®
- HP-UX IA64 (64-bit only)
- Linux x86
- zLinux
 - Red Hat Enterprise Linux
 - SUSE Linux Enterprise Server
- Solaris
- Windows

The following table lists the installation programs for ODBC/CLI and JDBC clients.

Table 6. ODBC/CLI and JDBC installation programs

Installation program	Operating system
cac111ax	AIX
cac111hpie64	HP-UX IA64
cac111lx	Linux
cac111zlx	zLinux
cac111so	Solaris
cac111wn.exe	Windows

To install the ODBC/CLI and JDBC clients:

Procedure

1. Log in as a user with administrator authority (user root on UNIX and Linux).
A non-root user can also install the client programs, but can update and uninstall only that user's instance.
2. Run the installation wizard.

Restriction:

If you do not use the default installation path, ensure that the directory you choose does not contain prior versions of the ODBC/CLI client.

The default installation paths are as follows:

Linux: /opt/ibm/isclassic111
UNIX: /opt/IBM/isclassic111
Windows: C:\Program Files\IBM\ISClassic111

If you install the client programs as a non-root user on UNIX or Linux, the default installation path is \$HOME/isclassic111.

Method	Steps
From CD	<p>Windows:</p> <ol style="list-style-type: none"> 1. Insert the Classic ODBC/CLI and JDBC client CD. 2. If autorun is enabled, the wizard will start automatically. If the wizard does not start, you can launch the wizard from a command prompt. <i>CD_drive:\cac111wn.exe</i> <p>UNIX and Linux:</p> <ol style="list-style-type: none"> 1. Insert and mount the Classic ODBC/CLI and JDBC client CD. 2. Start the wizard using the installation program that corresponds to the operating system as listed in Table 6 on page 32. <i>/mounted_volume/installation_program</i>
From ibm.com [®]	<ol style="list-style-type: none"> 1. On your system, create a temporary directory with at least 150 MB of free space. 2. Download the installation program that corresponds to the operating system into the temporary directory. 3. On Linux and UNIX systems, add execute permission to the installation program. 4. Start the wizard using the installation program that corresponds to the operating system as listed in Table 6 on page 32. <ul style="list-style-type: none"> • Windows: <i>Drive:\path_to_tempdir\cac95wn.exe</i> • UNIX and Linux: <i>cd</i> <i>/path_to_tempdir/installation_program</i> <i>chmod 755 ./cac111xx ./cac111xx</i>

Results

If you need to troubleshoot an install session, log information is available:

Windows:

install_path_log\install.log

UNIX, Linux:

install_path/_log/install.log

You can run the installation program with the following optional program arguments:

- help or -? for help (UNIX, Linux only)
- r to generate a response file
- f to input response file

Installing the ODBC and JDBC drivers from the command prompt

You can install the Classic ODBC and JDBC drivers without launching the graphic interface. The two alternate installation methods are called *silent* and *console*.

About this task

The console method is useful if you want to interact with the installation program but you do not have a system capable of displaying the graphic interface (for example, if you are accessing the system remotely in a secure shell).

The silent method, which is not interactive, is useful if you are accessing the systems remotely. The silent method can use settings that you provide in a response file, and enables you to run unattended installations or installations on multiple systems that require the same settings.

In the following steps, *installation_program* refers to one of the following programs:

Scenario	Program name
Installing IBM Classic ODBC/CLI and JDBC clients on a Windows system from CD:	<i>CD_drive:\cac111wn.exe</i>
Installing IBM Classic ODBC/CLI and JDBC clients on a Windows system from a download:	<i>Hard_Drive:\path_to_tempdir\cac111wn.exe</i>
Installing IBM Classic ODBC/CLI and JDBC clients on a Linux or UNIX system from CD:	<i>/mounted_cd_volume/cac111os</i>
Installing IBM Classic ODBC/CLI and JDBC clients on a Linux or UNIX system from a download:	<i>/path_to_tempdir/cac111os</i>

To install the ODBC and JDBC drivers from the command line:

Procedure

• Console method:

1. Run the installation program with the **-console** parameter.
 - ODBC/CLI and Java™ client for Windows, UNIX, or Linux:
installation_program -i console

• Silent method:

1. Optional: Create a response file to use for the silent installation.

You can create a response file by running the installation program in GUI or console mode with a form of the **-options** switch. You can edit the created response file in a text editor.

 - Using the installation wizard to create a response file only (for use in a later installation):
installation_program -options-template full_pathname_to_responsefile
 - Using the installation wizard to create a response file and install the product locally:
installation_program -options-record full_pathname_to_responsefile
2. Run the installation program with the **-silent** parameter.
 - ODBC/CLI and Java client for Windows, UNIX, or Linux:

```
installation_program -i silent
```

3. You can run the installation program with the following optional program arguments:

- help or -? for help
- r to generate a response file
- f to input response file

For example on Windows, UNIX or Linux:

```
installation_program -i console -r <path><properties.txt>
```

Results

If you need to troubleshoot an install session, log information is available:

Windows:

```
install_path\_log\install.log
```

UNIX, Linux:

```
install_path/_log/install.log
```

Uninstalling the Classic client components

Each Classic client component comes with an uninstall program. The uninstall program for the ODBC/CLI and JDBC clients can be run in interactive mode or silently.

Uninstalling the ODBC/CLI or JDBC client

You can uninstall the ODBC/CLI client and JDBC client by using the included uninstall facility.

About this task

To uninstall the ODBC/CLI or JDBC client:

Procedure

Run the uninstall wizard:

- **Windows:** Run the uninstall program. You can run it interactively or in silent mode. (The examples depict the default installation directory.)
 - **Interactive wizard:** Open the Add/Remove Programs Control panel, select IBM InfoSphere ODBC Administrator, and click **Change/Remove**.
 - **Interactive text-based console:**

```
cd C:\Program Files\IBM\ISClassic111\  
_uninst\uninstaller.exe -is:javaconsole -console
```
 - **Non-interactive or silent:**

```
cd C:\Program Files\IBM\ISClassic111\  
_uninst\uninstaller.exe -silent
```
- **Linux and UNIX:** As root user:
 1. Change to the directory where the ODBC/CLI or JDBC client is installed.
 2. Run the uninstall program. You can run it interactively or in silent mode. (The examples depict the default installation directory for Linux.)
 - **Interactive wizard:**

```
cd/opt/IBM/isclassic111/_uninst
./_uninst/uninstaller.bin
- Interactive text-based console:
cd/opt/IBM/isclassic111_uninst
./_uninst/uninstaller.bin -console
- Non-interactive or silent:
cd/opt/IBM/isclassic111_uninst
./_uninst/uninstaller.bin -silent
```

If you installed the client programs as a non-root user, substitute the following path for `/opt/ibm/isclassic111/_uninst`:

```
$HOME/isclassic111/_uninst
```

Results

If you need to troubleshoot an uninstall session, logs are available in the following locations:

Windows:

```
installation_path\_log\uninstall.log
```

Linux and UNIX:

```
installation_path/_log/uninstall.log
```

Notices

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

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS," without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

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

The following terms are trademarks or registered trademarks of other companies, and have been used at least once in this information:

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft 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.
- Linux is a registered trademark of Linus Torvalds 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, or service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies

and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



Printed in USA

GC19-4169-00

