SCLM Developer Toolkit

Installation and Customization Guide

Release 1

SC31-6899-01

SCLM Developer Toolkit

Installation and Customization Guide

Release 1

SC31-6899-01

Note!
Before using this information and the product it supports, be sure to read the general information under “Notices” on page 61.

Second Edition (November 2005)

This edition applies to IBM SCLM Developer Toolkit, Release 1, Program Number 5655M9900 and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

The IBM SCLM Developer Toolkit web site is at

`http://www.ibm.com/software/awdtools/sclmsuite/devtoolkit/`

The latest edition of this document is always available from the web site.

Contents

Figures

Tables

About This Document

This document contains the installation procedure for the IBM SCLM Developer Toolkit product, which combines standard z/OS® installation procedures with z/OS UNIX System Services and IBM z/OS HTTP server configuration.

From here on, the following names are used in this manual:
- The IBM z/OS HTTP server is called the "HTTP server".

Sometimes, "IBM SCLM Developer Toolkit" is shortened to "SCLMDT".

Who should use this document

This document is written for system programmers who are installing, configuring and administering the IBM SCLM Developer Toolkit product. Readers should be familiar with the z/OS UNIX System Services environment, Hierarchical File System (HFS) structure, Resource Access Control Facility (RACF®) profiles needed to support z/OS UNIX System Services and started tasks (or the equivalent for the installed security product), and the HTTP server.

It also contains information to be used by the administrator of any SCLM projects that are using the Java® and z/OS UNIX System Services component languages. These administrators also need to be familiar with the z/OS UNIX System Services environment and HFS structures, REXX Script, and the Java Compiler and SCLM project and language definitions.

Changes from the previous edition

Second edition

Minor changes have been made to this edition to clarify the steps required for installation. These changes include the new sections "Batch job considerations" on page xiv, "JAVA/J2EE ANT XML build skeletons" on page 27 and "J2EE application deployment" on page 27.

The changes are marked with revision bars.

Where to find more information

Where necessary, this document references information in other books, using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of z/OS, see *z/OS Information Roadmap*. Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 6:30 a.m. through 5:00 p.m. Mountain Time. You can use this number to:
- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates.

Hardcopy publications

Table 1. Hardcopy Publications

Short Title Used in This Document	Title of Publication	Order Number
HTTP Server Guide	*HTTP Server Planning, Installing and Using*	SC31-8690
z/OS UNIX System Services Planning	*z/OS UNIX System Services Planning*	GA22-7800
z/OS UNIX System Services Messages	*z/OS UNIX System Services Messages and Codes*	GA22-7807
z/OS UNIX System Services Commands	*z/OS UNIX System Services Command Reference*	GA22-7802
SCLM Project Manager's Guide	*z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide*	SC34-4817
SCLM Reference	*z/OS ISPF Software Configuration and Library Manager Reference*	SC34-4818

Softcopy publications

The z/OS library is available on the z/OS Collection Kit, SK2T-6700. This softcopy collection contains a set of z/OS and related unlicensed product books. The CD-ROM collection includes the IBM Library Reader™, a program that customers can use to read the softcopy books.

Softcopy z/OS publications are also available for Web browsing. PDF versions of the z/OS publications for viewing or printing using Adobe Acrobat Reader are available at this URL:

- http://www.ibm.com/servers/eserver/zseries/zos

Select "Library".

IBM Systems Center publications

IBM Systems Center produced Redbooks™ that can be helpful in setting up and using z/OS UNIX System Services. You can order these publications through the usual channels, or you can view them with a Web browser from this URL:

`http://www.redbooks.ibm.com`

These books have not been subjected to any formal review nor have they been checked for technical accuracy, but they represent current product understanding (at the time of their publication) and provide valuable information about a wide range of z/OS topics. You must order them separately. A selected list of these books follows:

Table 2. IBM Systems Center Publications

Title of Publication	Order Number	Comments
P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook	SG24-4757-01	Despite the title, it is oriented toward the system programmer, and describes considerations for the z/OS UNIX System Services environment.

Table 2. IBM Systems Center Publications (continued)

Title of Publication	Order Number	Comments
Debugging UNIX System Services, Lotus Domino, Novell Network Services	SG24-5613-00	Provides an overview of the z/OS UNIX System Services environment along with tips and suggestions for setup and problem analysis.
OS/390 e-business Infrastructure: IBM HTTP Server 5.1 - Customization and Usage	SG24-5613-00	Provides an overview of Web servers in general with specific details for the OS/390® server along with hints and tips for setup and customization.
e-business Enablement Cookbook for OS/390 Volumes 1, 2, and 3	SG24-5664-00 SG24-5981-00 SG24-5980-00	

Installation overview

This manual contains the installation procedure for all components of SCLM Developer Toolkit. The procedure is a combination of standard z/OS installation procedures, z/OS UNIX System Services file set up, and HTTP server configuration.

The manual is structured into four main parts
- Chapter 1, "Installing and customizing SCLM Developer Toolkit on z/OS," on page 1
- Chapter 2, "SCLM customization for the SCLM administrator," on page 23
- Chapter 3, "CRON-initiated Builds and Promotes," on page 37
- The appendixes, starting at page 41

TCP/IP considerations

When you are setting up the SCLM Developer Toolkit server, you also need to consider your site's installation of TCP/IP. SCLM Developer Toolkit uses an HTTP server and, therefore, needs to have the `TCPIP.DATA` file available to it. The z/OS UNIX System Services Planning guide documents where the system finds this file. However, if you use another method of defining the location of this file (such as the System Resolver), you need to add a `//SYSTCPD DD` card to your SCLM Developer Toolkit server job. A TCP port needs to be available and it is good practice to reserve the port number. To understand more about the System Resolver and TCPIP.DATA, see the following publications:
- z/OS UNIX System Services Planning
- z/OS IBM Communications Server: IP Configuration Guide

Security considerations

Each user that will use the SCLM Developer Toolkit must have a RACF OMVS, or equivalent, segment defined that specifies:
- A valid non-zero uid. The uid must be unique for each user.
- A valid home directory
- A valid shell

If these are not specified then the user will not be able to log on to the SCLM Developer Toolkit through the Eclipse IDE. If these are partially defined then unspecified errors may occur.

SMP/E installation

This manual does not cover the implementation aspects of SCLM Developer Toolkit. Rather, it is intended to guide the installer through a successful configuration of the product. The manual assumes that the System Modification Program/Extended (SMP/E) installation of SCLM Developer Toolkit has been completed. The SMP/E instructions for SCLM Developer Toolkit are in the IBM SCLM Developer Toolkit Program Directory, GI10-3352. Before you begin the SCLM Developer Toolkit installation, take note that the following actions were recommended for the SMP/E installation:
- The root HFS file system was made read only

- The SCLM Developer Toolkit directory was set up as a separate file system, mounted onto the root file system at `/usr/lpp/SCLMDT` (or whatever name you chose for your SCLM Developer Toolkit root directory).

These recommendations conform with those specified in the z/OS UNIX System Services Planning guide. See the section headed "Deciding How to Mount Your Root HFS for Execution" for full details.

Upon successful completion of the SMP/E install follow the directions in the next chapter to complete the install and customization on z/OS, and to download the Eclipse-based client on to the PC.

Note: It is recommended to have the BWBXXXXX modules reside in a dataset that is part of the LINKLIST. Alternatively this dataset may be added to the STEPLIB in the HTTP server (see "Step 4: Configure the SCLM Developer Toolkit HTTP Server" on page 4).

Batch job considerations

SCLM Developer Toolkit uses SDSF to retrieve job completion status and job output if requested. As not all customers are JES2 or have SDSF additional support has been put into SCLM Developer Toolkit to utilize the OUTPUT command. As shipped with z/OS the OUTPUT command will only let you retrieve job output that begins with the logged on userid. If you wish to utilize the OUTPUT facility fully then the supplied TSO/E exit IKJEFF53 may need to be modified to allow users to retrieve job output they own that does not begin with their userid. For more information on this exit see the z/OS TSO/E Customization guide.

Separate SCLM installation

This manual does not cover the implementation and loading of the SCLM product.

Chapter 1. Installing and customizing SCLM Developer Toolkit on z/OS

This chapter provides a list of the tasks required to install SCLM Developer Toolkit on you host z/OS system plus the tasks involved in installing the GUI component of the SCLM Developer Toolkit.

Steps 1 to 7 must be implemented by the z/OS Systems Programmer.

Step 8 is a PC-based installation that must be done by each user who wants to access SCLM using the SCLM Developer Toolkit.

For additional information on configuring SCLM Developer Toolkit for specific SCLM projects see Chapter 2, "SCLM customization for the SCLM administrator," on page 23. This chapter contains additional customization for:
- Site or SCLM project specific settings
- JAVA/J2EE language support

Step 1: Check z/OS software requirements

To successfully install SCLM Developer Toolkit, the following system requirements must be in place at your installation:

- z/OS V1.4 or above (preferably V1.5 or later) with the following PTFs applied:
  - ISPF PTF UA02085 (z/OS V1.4 only)
  - For customers wishing to store files with long filenames in SCLM, the PTF that addresses ISPF APAR OA11426 and provides support for long/short filename (long/short name) translation.
- z/OS UNIX System Services
- REXX runtime or REXX alternate libraries (`REXX.**.SEAGLPA`, or `REXX.**.SEAGALT`). REXX alternate is shipped with SCLM Developer Toolkit if required to be installed.
- ANT runtime installed in UNIX System Services, if performing JAVA/J2EE builds. (Download openly available from http://ant.apache.org/).

  See "Step 6: Install and customize ANT" on page 9.
- Java V1.3.1 (5655-D35) at service level dated May 10, 2003, or September 25, 2003 or later.

Step 2: Run the install JCL

1. Customize and run `BWBINST1`, which resides in the SBWBSAMP dataset.

   Follow the customization instructions within the member.

   This job performs the following tasks:
   - Creates CONFIG, LOGS and WORKAREA directories in the z/OS HFS at the directory specified by user customization.
   - Creates the PROJECT directory in the z/OS HFS under the CONFIG directory. For more information on the use of the PROJECT directory see "SITE and project-specific options" on page 32.
   - Copies the sample HTTP server configuration files from members `BWBHTTPC` and `BWBHTTPE` in the SBWBSAMP library to files `httpd.conf` and `httpd.env`

files in the CONFIG directory in the z/OS HFS install directory. These two files will require further tailoring. See "Step 4: Configure the SCLM Developer Toolkit HTTP Server" on page 4.

- Copies the Sample ISPF configuration table from member `BWBISPFC` in the SBWBSAMP library to file `ISPF.conf` in the CONFIG directory in the z/OS HFS install directory. This file will require further tailoring. See "Step 4: Configure the SCLM Developer Toolkit HTTP Server" on page 4.
- Copies the sample Translation configuration table from member `BWBTRANC` in the SBWBSAMP library to file `TRANSLATE.conf` in the CONFIG directory in the z/OS HFS install directory. This file will require further tailoring. See "Step 3: Customize the SCLM Developer Toolkit configuration files."

The recommended install directory for the configuration files is `/var/SCLMDT`. The `/var` directory must exist prior to this job being run.

SCLM Developer Toolkit users will require read and write access to the WORKAREA directory. By default users would require read and write access to `/var/SCLMDT/WORKAREA`. The WORKAREA is used for transfer of files, ASCII/EBCDIC conversions, and for JAVA/J2EE builds.

Temporary directories of format `/var/SCLMDT/WORKAREA/userid/*` are created during the use of Developer Toolkit. The following directories may be created under a directory of the users userid in the WORKAREA directory depending on the type of functions they are performing:

- `/EDIT`
- `/JobOutput`
- `/TRANSFER`
- `/VERSION`

Step 3: Customize the SCLM Developer Toolkit configuration files

The files `ISPF.conf` and `TRANSLATE.conf` reside at the default directory location `/var/SCLMDT/CONFIG` and may require further customization.

Customize the ISPF configuration file

You must customize the ISPF configuration table `ISPF.conf` that resides in the install directory to host site requirements for ISPF dataset allocation. The provided sample `ISPF.conf` has instructions to complete customization and enables the user site to:

- Include the minimum ISPF dataset allocations for SCLM Developer Toolkit operation.
- Add additional DDNAME file allocations or concatenate additional ISPF datasets.
- Execute a customer defined allocation exec to provide further dataset allocation by Project or Userid. A sample exec is provided in member `BWBISPF2` in the SBWBSAMP library.
- For batch operations ensure the ISPSLIB DD has the current ISPF/SCLM skeleton libraries allocated, as SCLM Developer Toolkit uses the standard allocated ISPF/SCLM skeletons.

Customize the TRANSLATE configuration file

Review the TRANSLATE configuration file `TRANSLATE.conf` that resides in the install directory. Follow the instructions contained within the sample if different ASCII/EBCDIC conversion codepages are required, besides the default of ASCII=ISO8859-1 and EBCDIC=IBM-1047.

Step 3: Customize the SCLM Developer Toolkit configuration files

The TRANSLATE.conf file provides keywords to determine how code is stored within SCLM. The provision to specify language types where files are binary transferred and stored, and also whether text based source remains in ASCII format rather than the default translation from ASCII to EBCDIC.

Additionally SCLM language types control whether longname files are converted to suitable valid short hostnames to store in SCLM. This long to short name mapping is controlled by the SCLM long/short name translate file.

Note: Default language types have been provided as a guide to use for determining long to shortname conversions, and/or BINARY transferred language types.

The following keywords are valid within the TRANSLATE.conf file:

Keyword	Description
CODEPAGE	Determines the ASCII and EBCDIC codepages to use in translation Format: • CODEPAGE ASCII = ISO8859-1 • CODEPAGE EBCDIC = IBM-1047 There must be a CODEPAGE keyword for both ASCII and EBCDIC for SCLM Developer Toolkit to determine how to convert files being transferred.
TRANLANG	This keyword determines which SCLM Language types require no ASCII/EBCDIC translation to the host (File will be binary transferred). If files were ASCII text they will remain in that ASCII codepage. ASCII/EBCDIC translation will take place for files by default. Format: • TRANLANG JAVABIN • TRANLANG DOC • TRANLANG JPEG In the above examples dummy Language Translators would be set up in SCLM for these languages. See Chapter 2, "SCLM customization for the SCLM administrator," on page 23 for more information on SCLM Language translators.
LONGLANG	This keyword determines which SCLM Language types require longname to shortname conversion. Longname to shortname translation implies the longname file on the Client (including directory package structure) will be mapped to a valid host member name of 8 characters and stored in SCLM using this translated host short name. Format: • LONGLANG JAVA • LONGLANG J2EEPART • LONGLANG DOC If the SCLM Language is not specified in the LONGLANG keyword then the Client file is assumed to already be in host short name format (8 characters or less) and is stored as is.

Usage scenario of the TRANSLATE configuration file

As stated above, the TRANSLATE configuration files control a number of things. Two of these are:

- Whether or not a part is going to be ASCII to EBCDIC translated.
- Whether a part is going to have its name translated from a long name to an 8 character short name.

To show how these two settings could be used to help in deciding how to initially set up this config file here is a usage scenario.

Usage scenario: You have a number of Word documents that you wish to store in SCLM. In this case files of this type cannot be edited on the mainframe. So there is no point in translating them to EBCDIC and they should just be stored in ASCII.

- Create an SCLM Language translator based on the sample `BWBTRANJ` in the SBWBSAMP library. You could call it BINARY (as this is how it will be stored), or if you want to be specific, call it DOC.
- As this file has a longname on the workstation, such as `InstallGuide.doc`, you need to make sure this is mapped to a generated short name in the SCLM PDS where you will store it. Therefore create a LONGLANG entry for the BINARY or DOC language, whatever you have called it.
- Add a TRANLANG entry for the BINARY or DOC language, whatever you have called it.
- When the Word document is checked out in SCLM it will be transferred down to the Eclipse workspace without any EBCDIC to ASCII conversion. Word will be started to work on the file. When changes are complete the Word document is checked back into SCLM and again no ASCII to EBCDIC translation takes place.

Step 4: Configure the SCLM Developer Toolkit HTTP Server

The SCLM Developer Toolkit interface uses a web server to communicate with the Developer Toolkit client in WebSphere®. This section describes the setup and customization of an HTTP server that is needed by the client to access SCLM on a z/OS host. It is recommended that this is a dedicated web server to support this interface, though optionally you may incorporate the required SCLM /HTTP configuration directives into an existing HTTP server.

See “Customizing an existing HTTP server for SCLM support” on page 6. By default the SCLM/HTTP server is configured to use port 80 though you may choose another suitable dedicated port during customization (1024 or higher as port numbers lower than this are reserved for internal systems use).

If you change the default port number it must be changed in the overriding server JCL.

The sample setup requires the end user to supply a valid z/OS userid and password when accessing the host system using this interface.

Note: For additional information on configuring IBM HTTP web servers, review these IBM manuals:

- HTTP Server Guide
- *OS/390 e-business Infrastructure: IBM HTTP Server 5.1 - Customization and Usage*

For the publication order numbers for these books, see “Where to find more information” on page ix.

The following sections outline the steps for customizing the supplied samples and starting the HTTP server.

By default the HTTP server configuration file and the environment file reside in the SCLM Developer Toolkit install directory. Optionally these files may be copied to another user directory or existing server configuration and environment files used. In all cases the HTTP server started task must be customized to reflect the appropriate directory.

HTTP server configuration file customization

Customize the sample HTTP configuration file `httpd.conf` (which resides in the install directory specified by install job `BWBINST1`) by following the instructions in the configuration file for the changes that are needed. The following directives need to be reviewed:

Table 3. httpd.conf customization

Directive	Description of change
Port	Leave as port 80 or change to a valid port number as specified in the HTTP server JCL. It is good practice to reserve the port number.
Protection and ServerId	Change `SCLMDTWB` to the name of the HTTP server job.
PidFile AccessLog ErrorLog	Change `/var/SCLMDT` to the appropriate path if a different path to the default was selected.
Pass and Exec Directives	Change `/var/SCLMDT` to the appropriate path if a different path to the default was selected. Change `/usr/lpp/SCLMDT` to the appropriate path of the bin install directory if a different path to the default was selected.

Non-standard codepage translation in SCLM Developer Toolkit	If users require different ASCII/EBCDIC codepage translation other than standard default (IBM-1047/ISO8859-1) the following parameters must be coded in the httpd.conf file for the HTTP server:
	`DefaultFsCp ebcdic-codepage`
	`DefaultNetCp ascii-codepage`
	
	For example, for Japanese translation the required codepages would be:
	`DefaultFsCp IBM-939`
	`DefaultNetCp IBM-932C`

HTTP server environment file customization

Customize the sample HTTP environment variables file `httpd.env` (which resides in the install directory specified by install job `BWBINST1`) by following the instructions in the environment file for the changes that are needed.

Table 4. httpd.env customization

Directive	Description of change
PATH	Ensure the PATH directive has the correct Java path directory

Table 4. httpd.env customization (continued)

Directive	Description of change
CGI_VCMPATH	This directive determines the HOME directory path where the configuration files reside. The default is: `CGI_VCMPATH=/var/SCLMDT`
CGI_TRANTABLE	This directive determines the name of the translate table used in short to long name translation. This VSAM file is discussed in "Step 5: Configure long/short name table VSAM file" on page 7. The default is: `CGI_TRANTABLE=SCLMDT.LSTRANS.FILE`

HTTP server JCL/STARTED TASK customization

Before the HTTP server can be submitted the following tasks must be performed:

1. Copy the sample batch job `BWBSRVR` from the installed sample library to a JCL Library or PROCLIB dataset and customize to your site specific standards and information by following the instructions in the sample.
2. Issue the CAPS OFF command to ensure that case sensitive values do not get changed to upper case.
3. It is recommended to make this HTTP server job a started task. But it can be run as a standalone job in order to test the HTTP server JCL.
4. If performing foreground Java builds it is recommended to use a region size of 128M in the HTTP server job.
5. RACF considerations:
   - Create a RACF OMVS segment for the userid assigned to the started task or if submitting the server as a job, the userid assigned to the job.
   - The HTTP server owning userid should have read access to the `BPX.SERVER` facility class.
   - The HTTP server owning userid requires execute access to the `/usr/lpp/internet/sbin` files and read/write access to the LOGS directory referenced in the `httpd.conf` file. This is defaulted to `/var/SCLMDT/LOGS`.
6. The default port to be used is 80. If you change this to a specific dedicated port you must also change the port number in the `httpd.conf` configuration file to match the port number in the started task JCL.
7. If the `BWBXXXXX` modules do not reside in the LINKLIST then edit the STEPLIB to specify the load library containing these modules. By default these are in the SBWBLOAD library.

   If the module `BWBTSOW` is included in this dataset then the dataset must be APF authorized.
8. Ensure a REXX/370 runtime environment on the host exists or alternatively use the REXX/370 Alternate Library supplied with SCLM Developer Toolkit.

   The REXX runtime library `REXX.**.SEAGLPA` (or if not installed, the alternate library `REXX.**.SEAGALT`) must be added to the STEPLIB in the HTTP server's JCL if they are not already defined as LINKLIST datasets.

   These datasets must be APF authorized.

Customizing an existing HTTP server for SCLM support

Follow the instructions below if you optionally choose to incorporate the SCLM Developer Toolkit support into an existing HTTP server.

Add the following pass/exec directives in the `httpd.conf` configuration file:

```
Pass    /J2EEPUT/               /var/SCLMDT/WORKAREA/*
Pass    /DWGET/                 /var/SCLMDT/WORKAREA/*
Pass    /DWTRANSFER/            /var/SCLMDT/WORKAREA/*
Pass    /BWBIVP.html            /usr/lpp/SCLMDT/bin/BWBIVP.html
Pass    /SCLMDW.html            /usr/lpp/SCLMDT/bin/SCLMDW.html
Pass    /DT*                    /usr/lpp/SCLMDT/bin/DT*
Exec    /BWBCALL                /usr/lpp/SCLMDT/bin/BWBCALL
Exec    /BWBIVP.cgi             /usr/lpp/SCLMDT/bin/BWBIVP.cgi
```

Figure 1. Pass and exec directives

Note: The following tailoring should be made to this example:

- Replace `/usr/lpp/SCLMDT` with your bin install directory.
- Replace `/var` with your Configuration/Workarea directory (may be the same as the bin directory).

Start the SCLM Developer Toolkit HTTP server

Start the web server by submitting the job or if a started task procedure enter the following command from the z/OS console:

```
/S server_proc_name.
```

(Ensure the procedure is a member in a PROCLIB dataset.) Check the HTTP server has successfully initialized. If the verbose trace is set on (-vv flag in Started Proc PARM) the server JOBLOG should contain the message:

```
AcceptExtended... Waiting for connection...
```

Step 5: Configure long/short name table VSAM file

SCLM Developer Toolkit provides the ability to store long name files (which are files with names greater than 8 bytes in mixed case) into SCLM. This is achieved through a VSAM file that contains the mapping of the long file name to the 8 character member name used in SCLM.

This facility is provided via a base ISPF/SCLM PTF that addresses APAR OA11426. This PTF must be applied before the long to short name translation can be used.

If this installation of SCLM Developer Toolkit is going to utilize this facility then the PTF providing the facility is installed and the VSAM file containing the mapping from long to short name can then be allocated. To do so, the following sample JCL, supplied with the PTF, can be modified and submitted.

Step 5: Configure long/short name table VSAM file

```
//FLM02LST JOB <JOB PARAMETERS>
//* ------------------------------------------------------------
//* ALLOCATION OF LONGNAME/SHORTNAME VSAM FILE
//*
//* THIS JOB ALLOCATES THE LONGNAME TO SHORTNAME TRANSLATE FILE.
//* THIS TRANSLATE FILE IS REQUIRED FOR THE FOLLOWING SCLM SUITE
//* PRODUCTS - SCLM DEVELOPER TOOLKIT AND SCLM ADMIN TOOLKIT.
//* THE ONE TRANSLATE FILE IS RECOMMENDED TO BE DEFINED AND USED
//* FOR ALL SCLM PROJECTS.
//*
//*
//*  CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.
//*  BEFORE USING THIS SAMPLE, YOU WILL HAVE TO MAKE THE
//*  FOLLOWING MODIFICATIONS:
//*
//*  1) ADD THE JOB PARAMETERS TO MEET YOUR SYSTEM REQUIREMENTS
//*
//*  2) CHANGE ALL REFERENCES OF HLQ.LSTRANS.FILE BELOW TO YOUR
//*     REQUIRED NAMING CONVENTION FOR THE SCLM TRANSLATE FILE.
//* ------------------------------------------------------------
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
  DELETE HLQ.LSTRANS.FILE
  SET MAXCC=0
  DEFINE CLUSTER(NAME(HLQ.LSTRANS.FILE)                      -
                 MANAGEMENTCLASS(STANDARD)                   -
                 RECSZ(58  2048)                             -
                 INDEXED                                     -
                 CYLINDERS(1 1)                              -
                 SHR(3,3)                                    -
                 KEYS (8 0))                                 -
  DATA(NAME(HLQ.LSTRANS.FILE.DATA))                          -
  INDEX(NAME(HLQ.LSTRANS.FILE.INDEX))
    /* DEFINE ALTERNATE INDEX WITH NONUNIQUE KEYS -> ESDS     */

    DEFINE ALTERNATEINDEX (                                  -
                 NAME(HLQ.LSTRANS.FILE.AIX)                  -
                 RELATE(HLQ.LSTRANS.FILE)                    -
                 RECORDSIZE(58  2048)                        -
                 CYLINDERS(1 1)                              -
                 KEYS(50 8)                                  -
                 NONUNIQUEKEY                                -
                 UPGRADE )                                   -
           DATA (                                            -
                 NAME(HLQ.LSTRANS.FILE.AIX.DATA) )           -
           INDEX (                                           -
                 NAME(HLQ.LSTRANS.FILE.AIX.INDEX) )
```

Figure 2. Sample Long/Short Translate VSAM file JCL (Part 1 of 2)

```
/*
//* ------------------------------------------------------------
//*  NOTE: THE FOLLOWING STEP WILL GET RC=4 DUE TO THE ALTERNATE
//*        INDEX BEING EMPTY. THE MESSAGES RETURNED ARE AS FOLLOWS.
//*
//*        IDC3300I  ERROR OPENING HLQ.LSTRANS.FILE
//*        IDC3351I ** VSAM OPEN RETURN CODE IS 100
//*        IDC0005I NUMBER OF RECORDS PROCESSED WAS 1
//*        IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4
//*
//* ------------------------------------------------------------
//IDCAM2   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//INITREC  DD *
INITREC1
/*
//SYSIN DD *
    REPRO INFILE(INITREC) -
          OUTDATASET(HLQ.LSTRANS.FILE)

    BLDINDEX IDS(HLQ.LSTRANS.FILE)                          -
             ODS(HLQ.LSTRANS.FILE.AIX)

    IF LASTCC = 0 THEN                                      -
             DEFINE PATH (                                    -
                      NAME(HLQ.LSTRANS.FILE.PATH)             -
                      PATHENTRY (HLQ.LSTRANS.FILE.AIX))
/*
```

Figure 2. Sample Long/Short Translate VSAM file JCL (Part 2 of 2)

For more information on the Long to Short name translation process see Appendix A, "Long/short name translation table," on page 41.

Step 6: Install and customize ANT

This step is required if you intend to use the JAVA/J2EE build support in SCLM.

ANT is openly available and can be downloaded from http://ant.apache.org/. ANT text files and scripts are distributed in ASCII format and require an ASCII/EBCDIC translation to run on z/OS in UNIX System Services. A sample translate script has been supplied in the SCLM Developer Toolkit SBWBSAMP library in sample member `BWBTRANT` and a sample copy job to copy the translate script into the appropriate ANT directory in sample member `BWBCPANT`. Follow the steps below to implement ANT on z/OS:

- Download latest ANT zip file into the z/OS HFS and unzip into the appropriate directory. (If the file is a TAR file, use the TAR extract command `tar -xf filename`, or if it is a zip file, use the Jar extract command `jar -xf filename`).
- Customize the install copy member `BWBCPANT` to include the ANT install directory and run the job to copy the translate script into the directory and translate (review the instructions contained within the sample member `BWBCPANT`).
- To check successful translation, locate a text file within the ANT directory such as the file `README` and browse.
- Change file permissions for all files under the ANT install directory to enable all users to read and execute.

  For example

  ```
  cd /u/antdirectory/ANT/apache-ant.1.6.2; chmod -R 755 *
  ```

Step 6: Install and customize ANT

- Before using ANT, set the z/OS UNIX System Services environment variables JAVA_HOME and ANT_HOME.
  - JAVA_HOME is required to point to the Java home directory, for example:

    ```
    JAVA_HOME=/usr/lpp/java/IBM/J1.3
    ```

  - ANT_HOME is required to point to the ANT install directory, for example:

    ```
    ANT_HOME=/u/antdirectory/ANT/apache-ant.1.6.2
    ```

  - ANT will look for an ANT configuration file in the directory /etc, so we recommend creating an ANT configuration file named ant.conf and adding the variables JAVA_HOME and ANT_HOME. That is, in file /etc/ant.conf set:

    ```
    JAVA_HOME=/usr/lpp/java/IBM/J1.3
    ANT_HOME=/u/antdirectory/ANT/apache-ant.1.6.2
    ```

    Note: The above directory paths are only sample directory paths. Ensure the correct directory paths are used.

    These variable may also be set in other ways:
    - Define them in the system wide profile (/etc/profile).
    - Define them in the users own .profile member in their home directory.
  - To test that the ANT initialization has been successful:
    - Add the ANT and Java bin directories to the environment variable PATH. This PATH variable may be added to your .profile or simply enter the following PATH definition below at the UNIX System Services command line.

      Example:

      ```
      export PATH=/u/antdirectory/ANT/apache-ant.1.6.2/bin:/u/javadirectory/IBM/J1.3/bin:$PATH
      ```

    - Run ANT to display the version.

      Example:

      ```
      ant -version
      ```

      This should display the ANT version if ANT is successfully installed.

Step 7: Run the IVP process to check correct install and customization

The HTTP server must be running and the IVP pass/exec directives configured in the httpd.conf file for successful verification processing.

From a browser, type the location URL address:

```
http://hostname:portnumber/BWBIVP.html
```

Where:

hostname Is the TCP/IP host name the HTTP server is running on.

port number Is the port used in the job and the httpd.conf file (default port 80).

If the HTTP server is running you will be prompted for a valid TSO userid and password for the system the web server is started on (Figure 3 on page 11).

Step 7: Run the IVP process to check correct install and customization

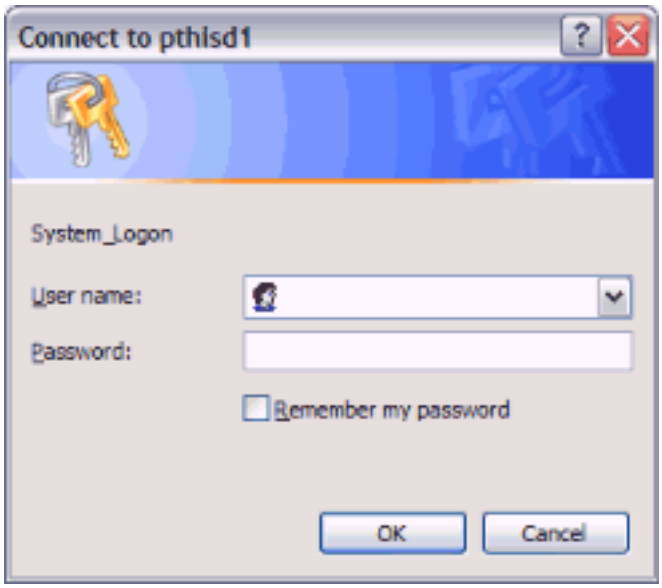

Figure 3. HTTP server logon prompt

Once you have entered your TSO userid and password the browser should initially display the html welcome screen (Figure 4).

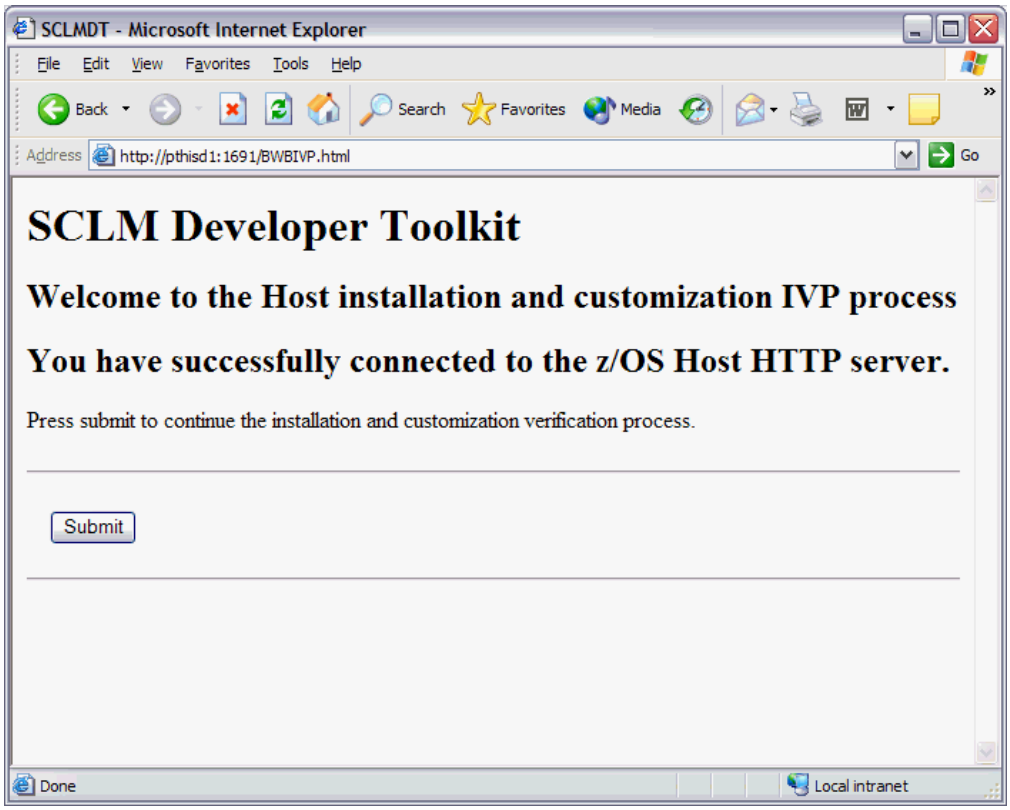

Figure 4. Host installation and customization welcome screen

If you fail to connect then check that:

- The HTTP server has successfully initialized.

Step 7: Run the IVP process to check correct install and customization

- The HFS mount point containing the SCLM Developer Toolkit installation is mounted.
- The `hostname:port` are correct (try pinging the Hostname).
- There are no firewall restrictions.
- The PASS directive in the `httpd.conf` file is set correctly:

```
Pass      /BWBIVP.html              /installdirectory/SCLMDT/bin/BWBIVP.html
```

Once you receive the welcome screen, continue with the IVP process, which checks and validates your install and customization process.

The sample screen (Figure 5) gives an example of how the validation responses may look.

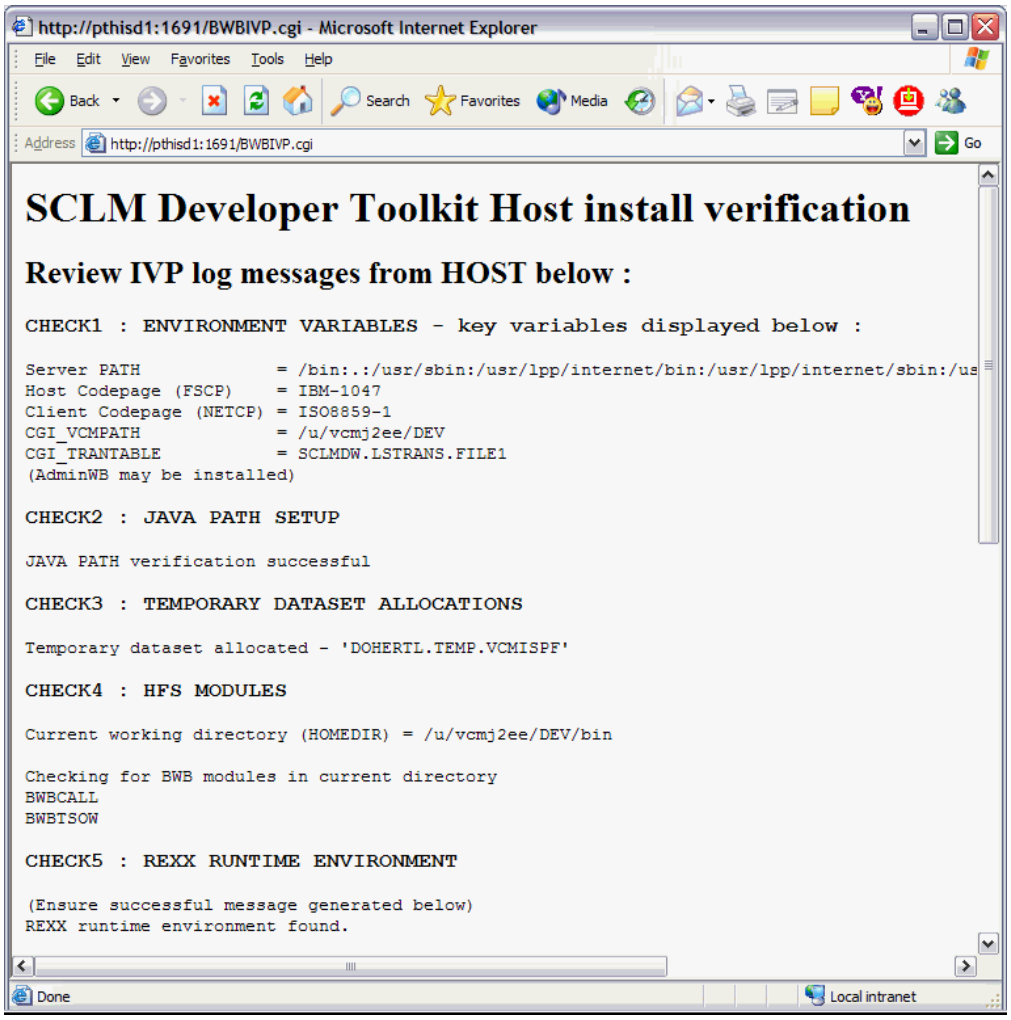

Figure 5. An example of validation responses (part 1)

Step 7: Run the IVP process to check correct install and customization

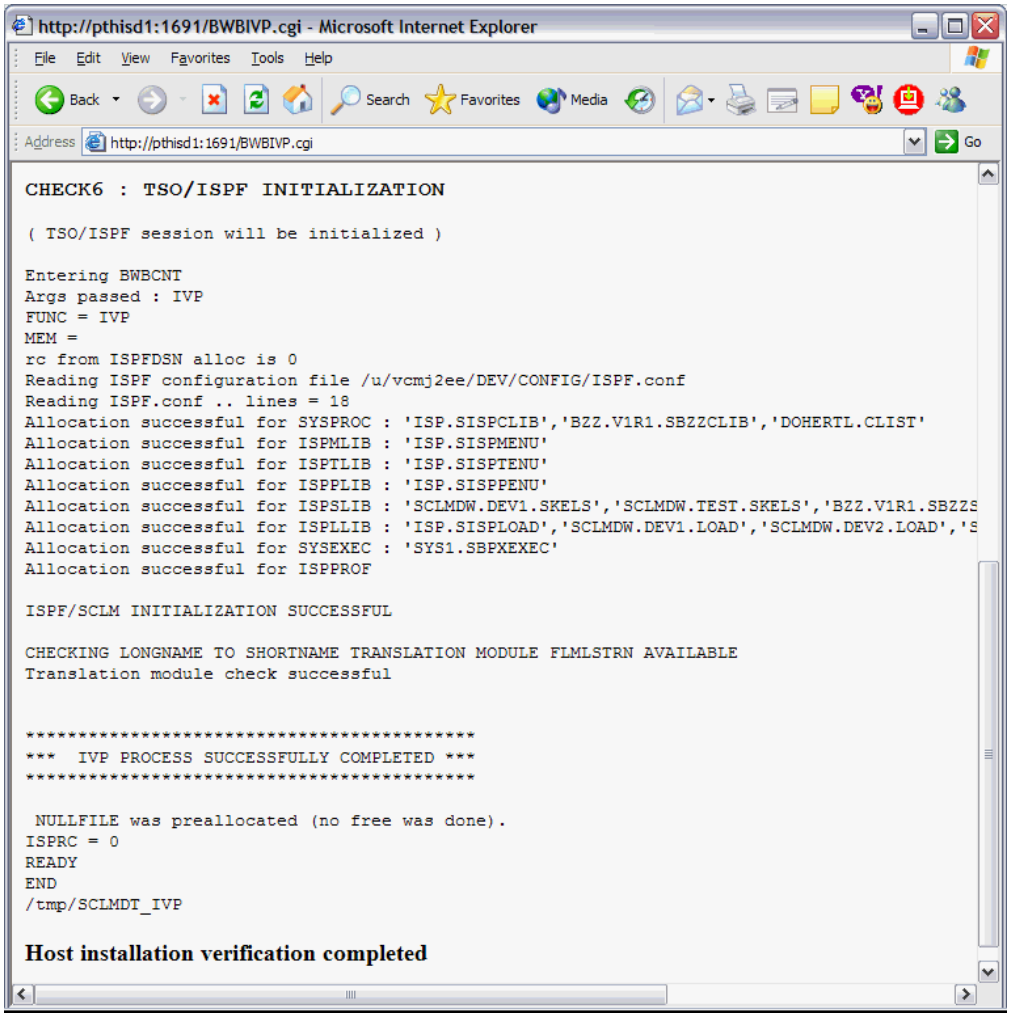

Figure 6. An example of validation responses (part 2)

Testing connection to the HTTP server

At any time the server may be tested to check if running without running the full IVP process check.

From a browser, type the location URL address:

```
http://hostname:portnumber/SCLMDW.html
```

Where:

hostname Is the TCP/IP host name the HTTP server is running on.

port number Is the port used in the job and the httpd.conf file (default port 80).

You are prompted for a valid userid and password for the system the web server is started on.

The browser should then display the following message:

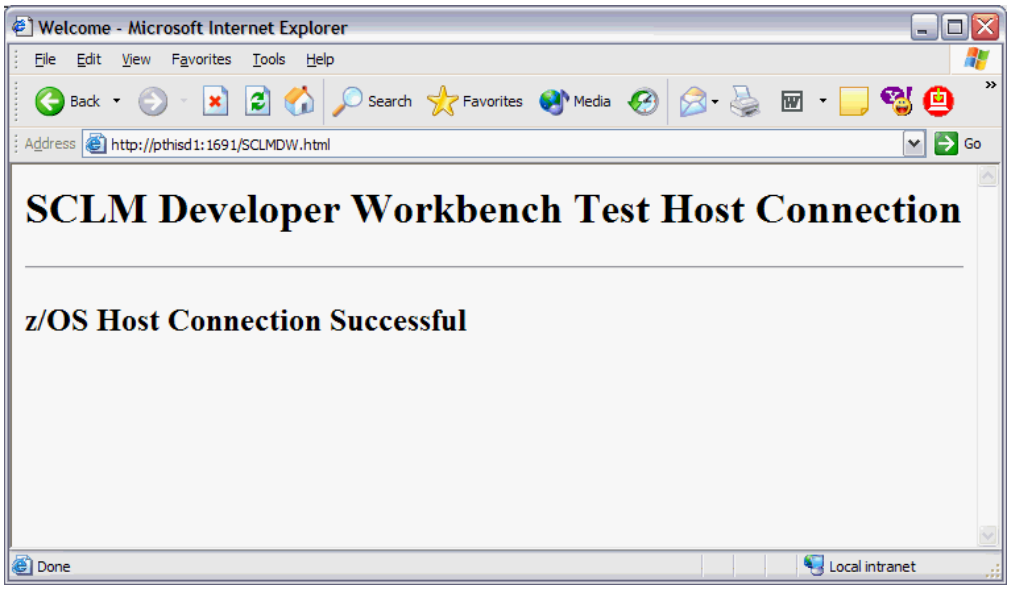

Figure 7. Server Connection successful message

Step 8: Download and install the Eclipse-based client on the users PC

The IBM SCLM Developer Toolkit client package installs the workstation code required to run the application using Install Shield.

This installation product consists of two main components, IBM WebSphere Studio Workbench (WSWB) and IBM SCLM Developer Toolkit plug-ins. The plug-ins require an Eclipse base in which to run. WSWB provides this. Alternatively you may have an existing Eclipse based product such WebSphere Application Developer into which the plug-ins can be installed.

The SCLM Developer Toolkit client package may be downloaded from the relevant z/OS HFS install directory via the Browser.

- Access the following URL from the browser (replace `hostname` and `port` with the relevant Host name and port number from the HTTP server customized in the previous steps):

  ```
  http://hostname:port/DTClient.html
  ```

  Where:

  hostname Is the TCP/IP host name the HTTP server is running on.

  port number Is the port used in the job and the `httpd.conf` file (default port 80).

  The browser should display the html download screen (Figure 8 on page 15).

Step 8: Download and install the Eclipse-based client on the users PC

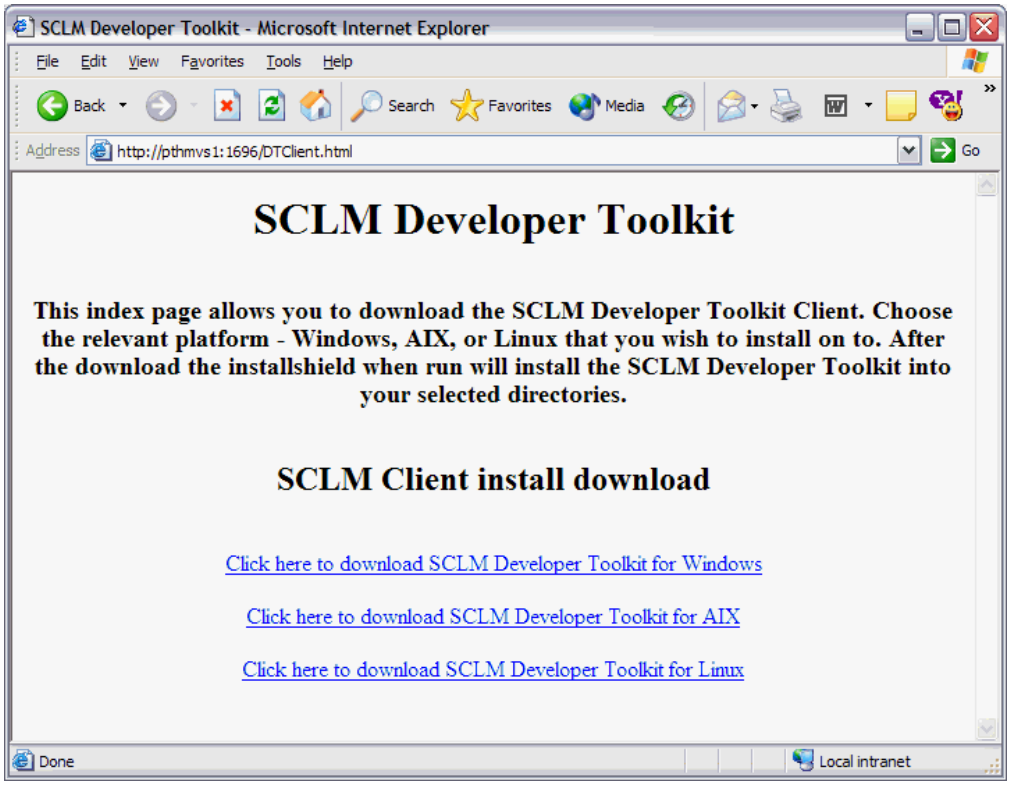

Figure 8. Client Download Index page

- From the list of client platforms as shown in (Figure 8), select the one is relevant for the operating system you are running. The Web Browser application will prompt you to open the file directly from the mainframe or save the file to a location on your machine. If you are saving the client install file on your machine, any location will suffice. If you are opening the file directly from the host the InstallShield process will start automatically. If you have saved the file onto your machine you must now run the install application. In Windows this can be done using the Run command from the Start menu or by double clicking on the file from Windows Explorer. In Linux™ and AIX™ the install program is run by opening a command line or terminal session and running the program as an ordinary UNIX command. Dependent on PATH information it may be necessary to have "./" prefix the command. User identity will be dependent on your system settings. Contact your systems administrator to determine under what user ID the program is to be installed under.
- The Welcome Screen is the first dialogue presented after running the installer

Step 8: Download and install the Eclipse-based client on the users PC

Figure 9. Install Shield Welcome Screen

- Clicking **Next** produces the Installation Overview dialogue.

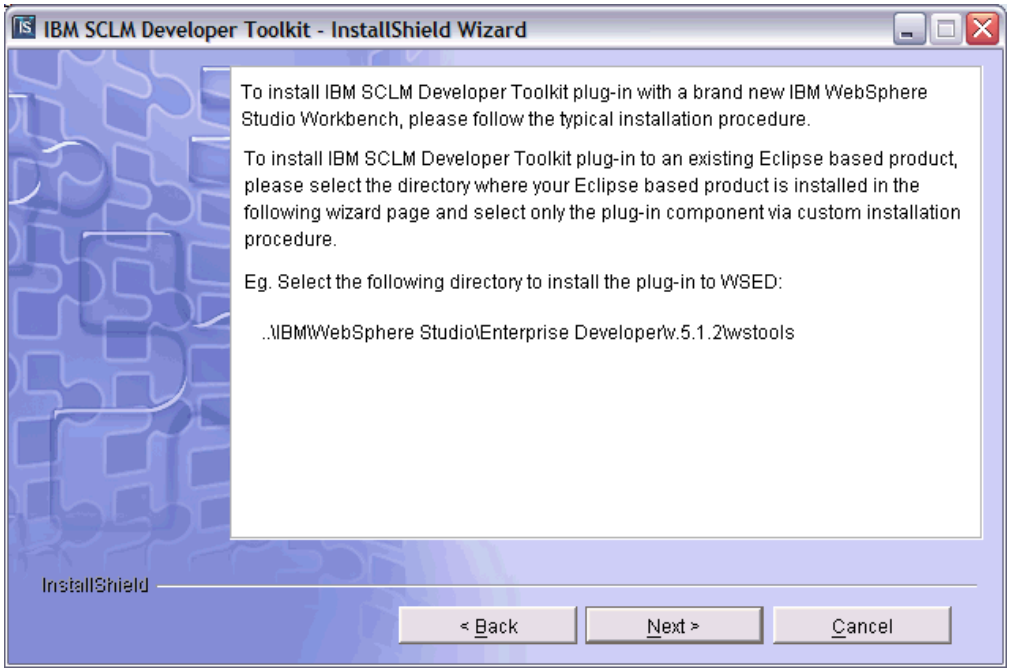

Figure 10. Install Shield Installation Overview Screen

- Two installation processes are available, Typical and Custom. These enable the user to install either the default configuration, IBM WebSphere Studio and the IBM SCLM Developer Toolkit plug-ins, or just the installation of the IBM SCLM Developer Toolkit plug-ins into an existing Eclipse base. Depending on which of

these installations will be selected will slightly vary the Install Shield screens the user will navigate through, plus the information required to be input by the user will also vary.

- If the user is installing onto a machine that currently does not have Eclipse installed then the Typical installation should be selected. The location of the installation directory must be provided as detailed in Figure 11. This is the location you wish to install the WSWB version of Eclipse plus the SCLM Developer Toolkit plug-ins.

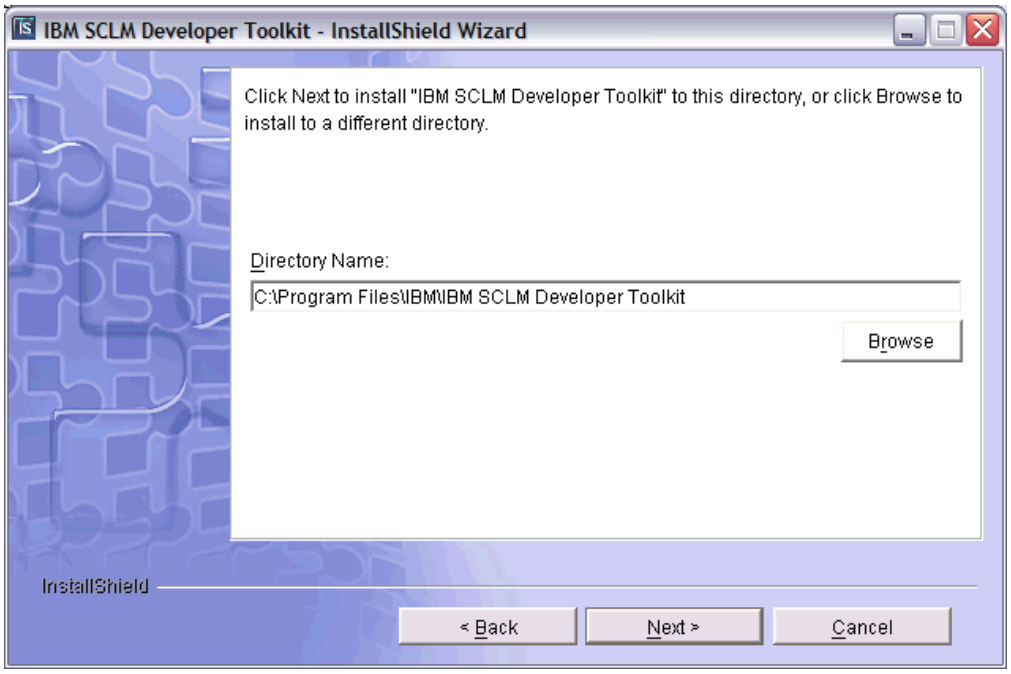

Figure 11. Install Shield Installation Directory Screen (New Eclipse)

- Clicking **Next** will display the setup choice screen (Figure 12 on page 18). Clicking **Typical** will select WebSphere Studio Workbench and the required plug-ins for installation into the prescribed directory.

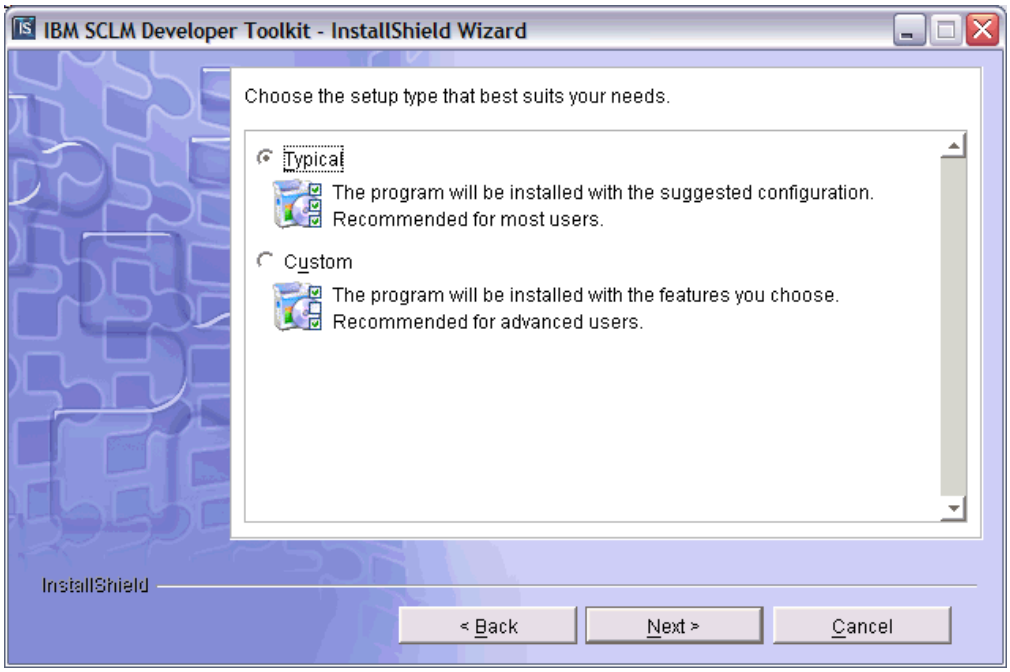

Figure 12. Install Shield setup type Screen (New Eclipse)

- A dialogue, shown in Figure 13, is presented confirming the details of the components to be installed and location.

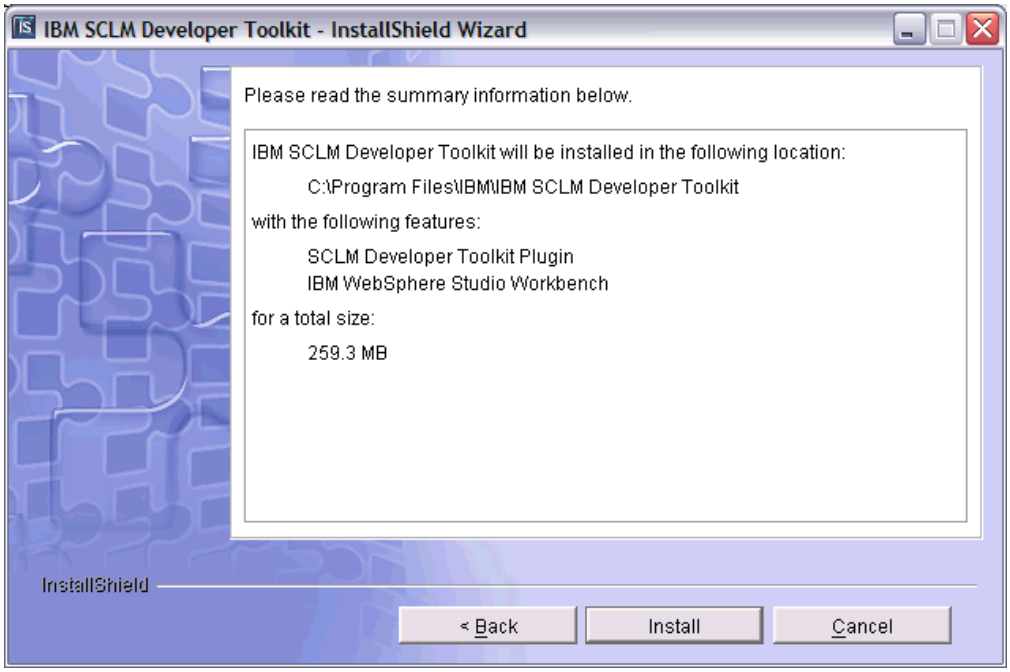

Figure 13. Install Shield Components and Locations Screen (New Eclipse)

- Clicking **Install** will initiate the installation process.
- If the user is installing onto a machine that currently has an Eclipse installed, such as WebSphere Application/Enterprise Developer then the Custom installation should be selected, as you can select the components you wish to install. The location of the installation directory must be provided as detailed in

Figure 14. This is the location of the existing Eclipse product. Normally the `/wstools/` directory is selected as there is an Eclipse directory there that the user should install the plug-ins into. Contact your systems administrator to confirm what the correct location should be, as it may vary between WebSphere versions.

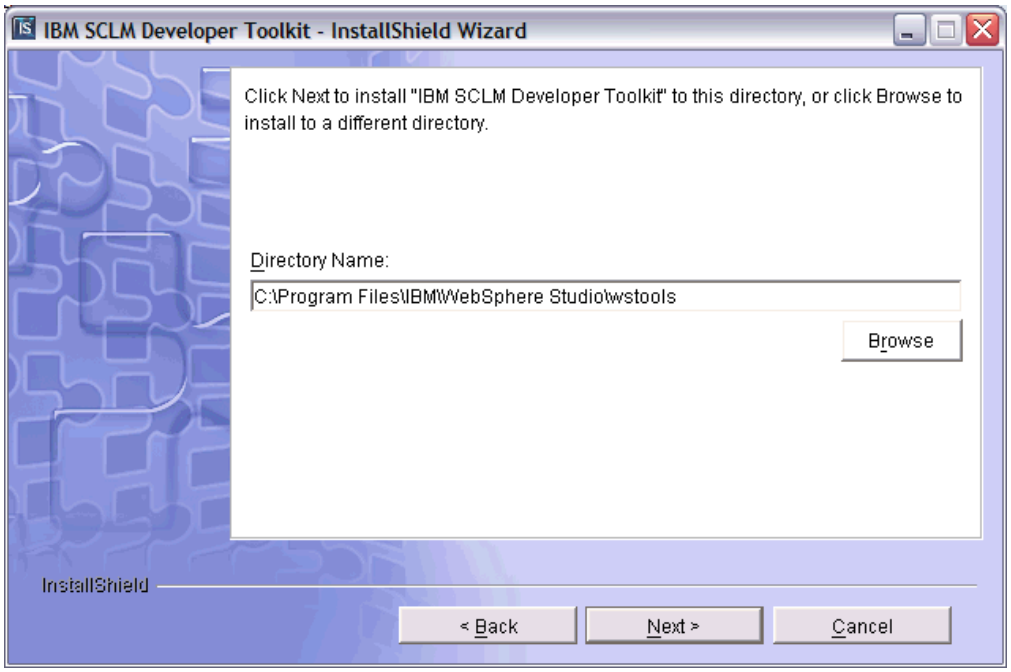

Figure 14. Install Shield Installation Directory Screen (Existing Eclipse)

- Clicking **Next** will display the setup choice screen (Figure 15 on page 20). Clicking **Custom** will allow the user to select which components need to be installed.

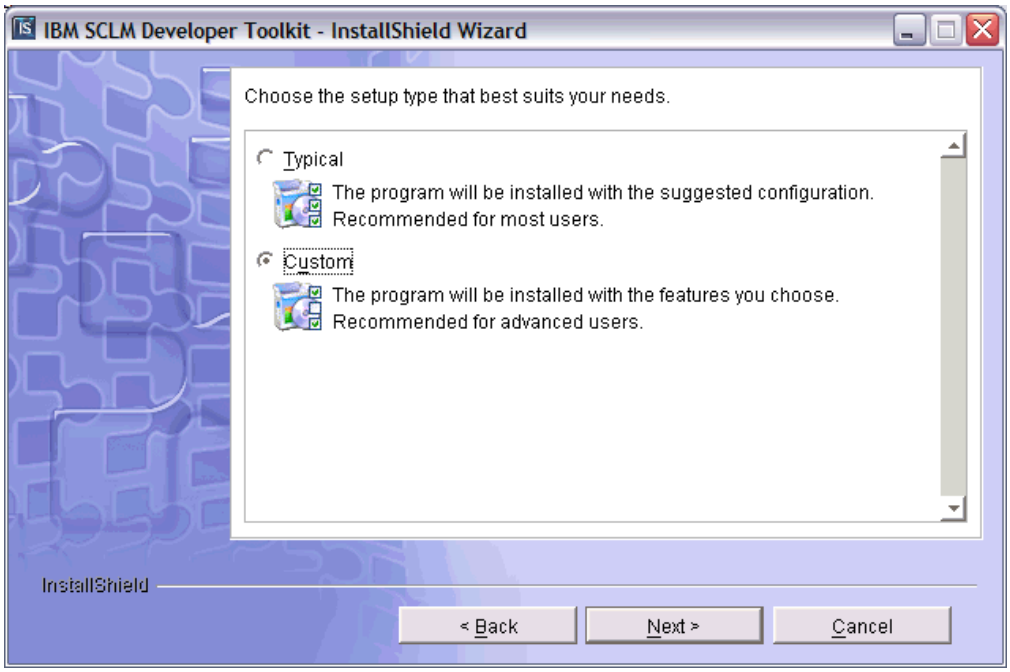

Figure 15. Install Shield setup type Screen (Existing Eclipse)

- As installation is into an existing Eclipse, only the SCLM Developer Toolkit Plug-in is selected.

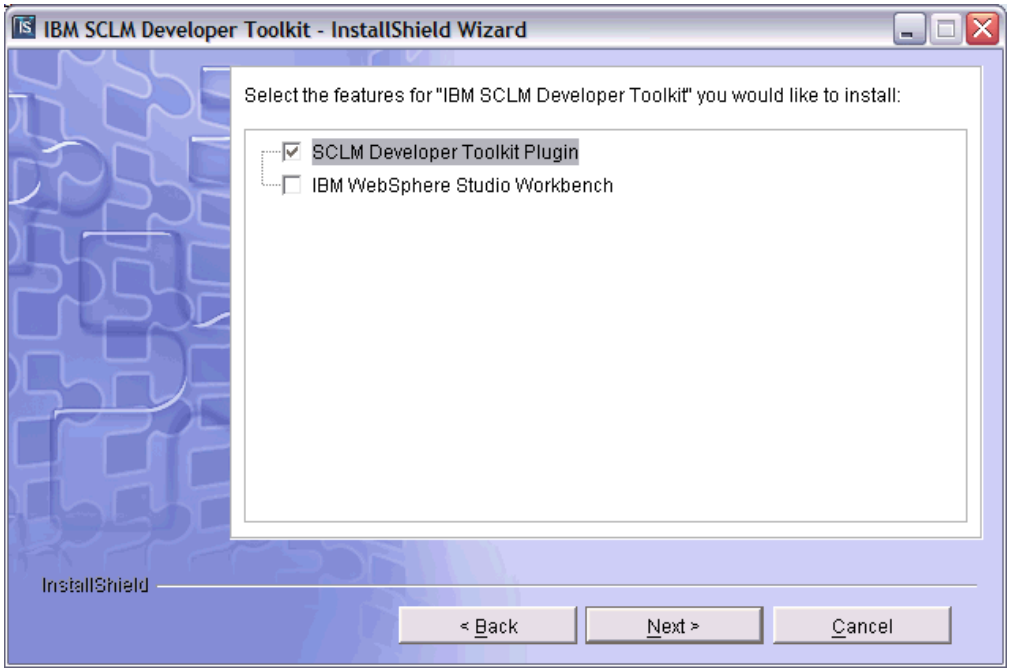

Figure 16. Install Shield Selected Components Screen (Existing Eclipse)

- Check the items to be installed and click **Next**. A dialogue, shown in Figure 17 on page 21 is presented confirming the details of the components to be installed and location.

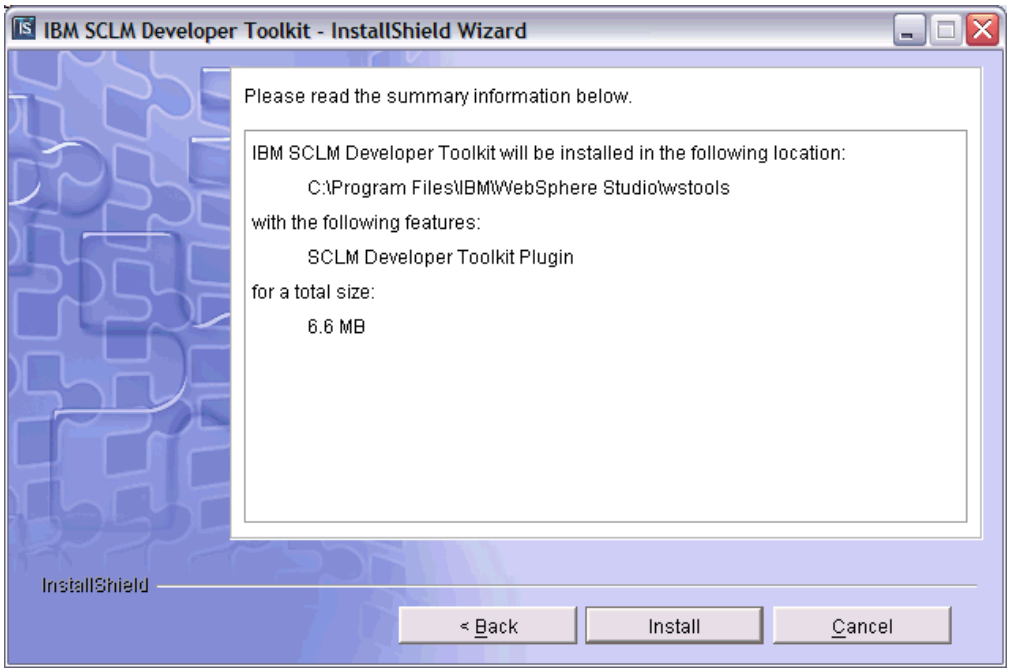

Figure 17. Install Shield Components and Locations Screen (Existing Eclipse)

- Clicking **Install** will initiate the installation process.
- In both the case of installing into an existing Eclipse and creating a new Eclipse installation a conformation screen is shown once installation is complete.

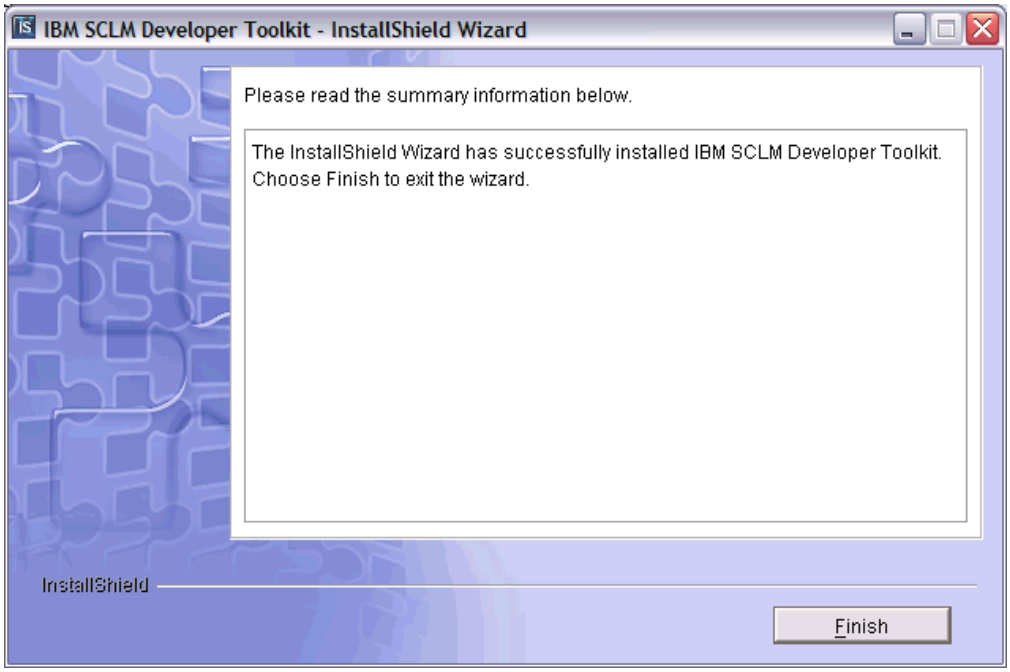

Figure 18. Install Shield Install Summary Screen

If the user has installed a new Eclipse installation, the application is launched by running the `eclipse.exe` file from the installation directory. If the plug-ins have been installed into an existing Eclipse installation then the user should run the Eclipse product as normal.

Step 8: Download and install the Eclipse-based client on the users PC

In either event, when you run the product for the first time you will be prompted to confirm the installation of the new plug-ins. Confirm this. Eclipse will then request that the application needs to be re-started. Accept this. Once Eclipse has re-started the installation is complete. To confirm the installation of the plug-ins, in the Eclipse IDE go to **Help -> About Eclipse -> Plug-in Details**. This will detail all of the installed plug-ins. There should be entries for IBM SCLM Connector, IBM SCLM Developer Toolkit and IBM SCLM Developer On-line Help.

The product is now installed.

Chapter 2. SCLM customization for the SCLM administrator

Language translators for JAVA/J2EE support

SCLM Developer Toolkit requires four new language translators defined in SCLM for JAVA/J2EE support. These language translators are shipped in the SBWBSAMP members as shown below:

Sample Translator	Description
BWBTRANJ	Sample default member translator. No parsing. Similar to SCLM FLM@TEXT.
BWBTRAN1	Sample Java language translator.
BWBTRAN2	Sample Java language translator incorporating ANT (for multiple Java compiles and Jar builds).
BWBTRAN3	Sample J2EE language translator for J2EE support using ANT (supports Jar, WAR, EAR files).

Figure 19. Sample translators

The SCLM Administrator will need to generate these samples into the PROJDEFS.LOAD library for each SCLM project where Java support is required. These translators are required to be added/compiled in the Project Definition.

A sample Project definition for JAVA/J2EE projects and Host components is provided in sample BWBSCLM.

The LOADLIB dataset containing the BWBXXXXX modules must be included in the ISPF ISPLLIB concatenation to access the JAVA/J2EE language translator modules. The ISPLLIB concatenation is customized in the configuration file ISPF.conf.

SCLM DATASETS for JAVA/J2EE:

It is recommended to create SCLM target source datasets of RECFM=VB, LRECL=1024 for JAVA/J2EE source being stored in SCLM from the Toolkit client to cater for long record types. The editors on the Eclipse-based client create files of variable record length and to maintain integrity the Host target datasets in SCLM should also be of RECFM=VB. Using Fixed record length datasets (RECFM=FB) will result in imported members having white spaces appended to end of record.

JAVA/J2EE build summary

Here is a summary of the process that occurs for Java and J2EE builds using the supplied translators.

Java build

- Compilation of all Java Source into output classes.
- Classes stored in SCLM and long/short name stored in Translate tables.

- Optional Jar created (contains Classes and may contain other Java project components such as XML/HTML etc in a packaged structure).
- Jar objects stored in SCLM and long/short name stored in Translate table.
- Jar structure determined by the archdef used. The longnames associated with the members in the archdef determine the Jar packaging format.

J2EE build

- Compilation of all Java Source into output classes.
- Classes stored in SCLM and long/short name stored in Translate tables.
- Optional EJB Jar created (contains Classes and may contain other Java project components such as XML/HTML/JSP etc in a packaged structure).
- Optional web WAR file created based on J2EE web.xml file in J2EE project and stored in SCLM as above.
- Optional EAR file created for deployment based on application.xml in J2EE project and stored in SCLM as above.
- All listing outputs go to J2EELIST.

SCLM language definitions

The sample translators define the following languages:

Translator	Description
J2EEPART	Language type that specifies a JAVA/J2EE component and defined by sample BWBTRANJ. No particular parsing occurs on build of this language type. Non-Java source or J2EE components that require ASCII/EBCDIC language conversion may be generically slotted under this language type if no particular build parsing is required (for example html, xml, definition tables). Optionally language type of TEXT may be used.
J2EEBIN	Language type that specifies JAVA/J2EE Binary or ASCII stored component and defined by sample BWBTRANJ. No particular parsing occurs on build of this language type. JAVA/J2EE binary files and text files that wish to be stored as ASCII may be generically slotted under this language type if no particular build parsing is required.
JAVA	Language type for Java source and defined by sample `BWBTRAN1`. The Java translator determines what type of build has been issued against Java source. • Scenario 1: Build issued against individual Java program. The Java translator compiles source into output CLASSES. Class is stored in SCLM. Javac compile output is stored in type JAVALIST. Any classpath dependencies may be satisfied by storing dependent Jars in the classpath directory specified in $GLOBAL member parameter `CLASSPATH_JARS`. • Scenario 2: Build against ARCHDEF (JAVABLD script which calls ARCHDEF) leaves the ANT script specified to do the build. The Java translator itself, when invoked by the ARCHDEF, just copies the output classes/Jar into SCLM. An antbuild summary file is stored in JAVALIST. Individual Java components have an output table stored in JAVALIST.
JAVABIN	Language type that is similar as above to Java but is used when storing Java source as ASCII in SCLM.
JAVABLD	Language type that specifies an ARCHDEF generated build. Generally an ARCHDEF itself is built against, but in the JAVA/J2EE environments a JAVABLD script is built against, which invokes a specified ARCHDEF to be built. The script of Language Type JAVABLD specifies an ARCHDEF name, ANTXML script to perform bulk builds, and other field tags to facilitate the antxml Build. The Jar created formats the contents exactly as specified in the ARCHDEF structure itself.
J2EEBLD	Language type that is similar as above to JAVABLD but for the full J2EE component structure resulting in additional objects such as WAR and EAR files being created.

Note: All objects such as Jar, WAR, and EAR have their internal zipped source parts in ASCII to distribute to all platforms.

Figure 20. Sample SCLM language translators

SCLM types

There are a number of SCLM types that need to be created for JAVA/J2EE support. Some of these types are mandatory types and must be created for JAVA/J2EE support to function.

Type	Description
J2EEBLD	This SCLM Type is required to be setup for both Java and J2EE build processes. The J2EEBLD type contains: • JAVABLD and J2EEBLD build scripts used to drive the ANT build process. • Java and J2EE ANTXML scripts to be invoked for bulk builds. **Note:** Sample Java and J2EE `ANTXML` scripts are supplied. Generally these scripts require little or no user customization. Site and user dependent variables are customized in the JAVABLD/J2EEBLD scripts themselves to override default ANTXML variables. (For more information, see "JAVA/J2EE ANT XML build skeletons" on page 27.) Contains a $GLOBAL member which is required for both Java and J2EE builds (see "$GLOBAL member" on page 31).
ARCHDEF	Contains Java and J2EE archdef members. JAVABLD Jars are built according to the archdef layout. For JAVA/J2EE processing the archdef type must be named archdef. The longname parts in each archdef member outline the JAVA/J2EE project structure. The archdef for a given project may be dynamically created from the client when migrating in new projects or updated when adding new parts. The JAVA/J2EE BUILD functionality differs from normal SCLM HOST component builds in that the archdef is not what is built against but the Build scripts in Type J2EEBLD which reference the archdef. These build scripts overlay project dependent build information and invoke the archdef and the ANT XML script specified.
JAVALIST	This SCLM Type is required to be setup for the Java build process. The JAVALIST type contains listing outputs from JAVABLD builds.
J2EELIST	This SCLM Type is required to be setup for the J2EE build process. The J2EELIST type contains listing outputs from J2EEBLD builds.
JAVACLAS	This SCLM Type is required to be setup for both Java and J2EE build processes. The JAVACLAS type contains output Class files from builds associated with the JAVA, JAVABLD and J2EEBLD language types.
JAVAJAR	This SCLM Type is required to be setup for Java builds (language type JAVA or JAVABLD). The JAVAJAR type contains Jar output from builds associated with the JAVA and JAVABLD language types.
J2EEJAR	This SCLM Type is required to be set up for J2EE builds (language type J2EEBLD). The J2EEJAR type contains Jar output from builds associated with the J2EEBLD language type.
J2EEWAR	This SCLM Type is required to be setup for the J2EE build process. The J2EEWAR type contains WAR output from builds associated with the J2EEBLD language type.
J2EEEAR	This SCLM Type is required to be setup for the J2EE build process. The J2EEEAR type contains EAR output from builds associated with the J2EEBLD language type.

Figure 21. SCLM types

JAVA/J2EE ANT XML build skeletons

Below is a list of sample ANT build skeletons which are provided in the SBWBSAMP library. These sample members can be copied into SCLM type J2EEBLD in the SCLM hierarchy to be referenced and used by the JAVA/J2EE build scripts.

The J2EE build skeletons work generically for standard J2EE projects and there is a separate sample for Enterprise javabeans (EJB) creating an EJB JAR, WEB application creating a web archive file (WAR), and assembly of an Enterprise archive file (EAR) and deployment sample. Be aware however that some J2EE projects may not fit the standard model and some customization of the supplied ANT XML skeletons may be required.

Note: JAVA/J2EE build scripts can be generated via the Developer toolkit client application. These build scripts use a referenced ANT XML skeleton (as below) and an archdef in the JAVA/J2EE build process.

A detailed description of build scripts, ANT skeletons and examples on JAVA/J2EE build processing is contained in the SCLM Developer Toolkit user guide supplied with the client plug-in.

BWBJAVAA Sample ANT XML JAVA build skeleton

This ANT skeleton would be used by a Java build script to compile multiple java programs and optionally create a Java Archive (JAR) file which has a structure determined by a specified archdef.

BWBEJBA Sample ANT XML J2EE EJB build skeleton

This ANT skeleton would be used by a J2EE build script to compile/build an EJB project which would usually create an EJB JAR which has a structure determined by a specified archdef.

BWBWEBA Sample ANT XML J2EE WEB build skeleton

This ANT skeleton would be used by a J2EE build script to compile/build a WEB project which would usually create a WEB Archive (WAR) file.

BWBEARA Sample ANT XML J2EE EAR assemble skeleton

This ANT skeleton would be used by a J2EE build script as an assemble process in preparation for J2EE application deployment. The process produces Enterprise Archive (EAR) files which can be deployed on to a web application server such as WebSphere application server.

BWBDEPLA Sample ANT XML DEPLOY skeleton

This ANT skeleton would be used by a J2EE build script to deploy an EAR file on to a web application server such as WebSphere application server residing on z/OS.

J2EE application deployment

IBM SCLM Developer Toolkit uses the IBM WebSphere Application Server (WAS) wsadmin tool to deploy J2EE applications to WAS running on z/OS. The wsadmin tool requires a JACL script to guide the deployment process. Hence the JACL script must be installed under Unix System Services before the deployment process can be invoked.

The JACL script must be copied into the HFS directory on z/OS. It is recommended to store the JACL script as `/var/SCLMDT/CONFIG/scripts/deploy.jacl` (where `/var/SCLMDT` is the default install directory for SCLM Developer Toolkit). The wsadmin tool expects the JACL script to be in ASCII format. A sample job is provided in the SCLM DT sample library which will copy the provided JACL script into the selected directory in ASCII format.

Note:
The JACL samples `BWBJACL` and `BWBJACLJ` were delivered in PTF UA22442. Before continuing with the procedure, please ensure that this PTF has been applied to SCLM Developer Toolkit if these samples are required.

To copy the sample JACL script `BWBJACL` (resides in the SBWBSAMP sample library) into a HFS directory in ASCII format perform the following:

- Review the instructions contained in the sample job `BWBJACLJ` and customize accordingly. `BWBJACLJ` can be found in the SCLM Developer Toolkit SBWBSAMP library.
- Submit the job `BWBJACLJ`. This will convert into ASCII sample `BWBJACL` and copy into the customized directory as file name `deploy.jacl`.

The Client plug-in must be aware of the directory locations where the wsadmin tool (`wsadmin.sh`) and the JACL script are installed under Unix System Services. Both locations can be configured in the preference page under **Team -> SCLM Preferences -> Build Script Options**. The Client plug-in is used to generate a deployment script which can then be built against. (The deployment process is triggered by an SCLM build of the deployment script which resides in SCLM type J2EEBLD).

A copy of the JACL sample script to be used is included below and should require no customization.

```
#--------------------------------------------------------------------
#
#  IBM SCLM Developer Toolkit JACL script for deployment
#
#--------------------------------------------------------------------

proc ex1 {args} {

   #--------------------------------------------------------------
   # set arguments
   #--------------------------------------------------------------

   set app        [lindex $args 0]
   set appName    [lindex $args 1]
   set cellName   [lindex $args 2]
   set nodeName   [lindex $args 3]
   set serverName [lindex $args 4]

   #--------------------------------------------------------------
   # set up globals
   #--------------------------------------------------------------
   global AdminConfig
   global AdminControl
   global AdminApp


   #--------------------------------------------------------------
   #     -- was a earfile name supplied
   #--------------------------------------------------------------
   if {[llength $app] == 0} {
      puts "deploy: Error -- No application specified."
      return
   }

   #--------------------------------------------------------------
   #     -- was the appname supplied
   #--------------------------------------------------------------
   if {[llength $appName] == 0} {
      puts "deploy: Error -- Application name not specified."
      return
```

```
   }

   #-------------------------------------------------------------
   # Uninstall the exiting application
   #-------------------------------------------------------------
   puts "Checking for installed applications:"

   set foundApplication "false"
   set application_list [$AdminApp list]
   # Get a list of application currently installed on the server.
   foreach application $application_list {
    # If the application is already installed on the server,
    # then mark the flag as installed already.
    if {$application == $appName} {
     set foundApplication "true"
    }
   }

   if {$foundApplication == "true"} {
    puts "The application is installed already. Uninstalling..."
    $AdminApp uninstall $appName
    $AdminConfig save
   }

   #-------------------------------------------------------------
   # Setup options for the deployment
   # Additional options can be added here as required
   # For Example:
   #  lappend app_options -update
   #  lappend app_options -appname MyAppName
   #  lappend app_options -contextroot MyAppName
   #  lappend app_options -preCompileJSPs
   #  lappend app_options -defaultbinding.force
   # for a full list of options please use the AdminApp command
   # wsadmin [return]
   # $AdminApp options            - generic options
   #      or
   # $AdminApp options MyApp.ear  - valid options for your ear file
   # lappend app_options -node WXP-KEFA25B
   #-------------------------------------------------------------
   puts "deploy: installing the application"

   set app_options [list -server $serverName]
   lappend app_options -node $nodeName
   lappend app_options -verbose

   #-------------------------------------------------------------
   # Install the application onto the server
   #-------------------------------------------------------------
   $AdminApp install $app $app_options

   #-------------------------------------------------------------
   # Save all the changes
   #-------------------------------------------------------------
   puts "deploy: saving the configuration"
   $AdminConfig save


   #-------------------------------------------------------------
   # Start the installed application
   #-------------------------------------------------------------
   puts "Starting the application..."
   set appManager [$AdminControl queryNames cell=$cellName,node=$nodeName,type=ApplicationManager,process=$serverName,*]

   $AdminControl invoke $appManager startApplication $appName
   puts "Started the application successfully..."

   puts "deploy: done."
}

#-------------------------------------------------------------
# Main
#-------------------------------------------------------------
if { !($argc == 5) } {
   puts "deploy: This script requires 5 parameter: ear file name, application name, cell name, node name and server name"
   puts "e.g.:     deploy  /WebSphere/AppServer/installableApps/jmsample.ear myapp1 myCell myNode myServer"
} else {
   set application      [lindex $argv 0]
   set appName          [lindex $argv 1]
   set cellName         [lindex $argv 2]
   set nodeName         [lindex $argv 3]
   set serverName       [lindex $argv 4]

   ex1 $application $appName $cellName $nodeName $serverName
}
```

ASCII or EBCDIC storage options

Source files transferred from the SCLM Developer Toolkit plug-in may be stored in SCLM as either ASCII or EBCDIC.

Generally all source in SCLM is stored in EBCDIC to be viewed and edited directly from ISPF/SCLM on z/OS. Users who do not wish to browse or edit code directly

from the host may wish to store code directly (that is, as binary transferred) where source will be stored in SCLM using the original client's ASCII/UNICODE codepage. This does have some performance benefits for large projects being stored and imported from SCLM and for JAVA/J2EE builds as an ASCII to EBCDIC translation will not be performed.

SCLM Developer Toolkit determines if a file is binary transferred or if an ASCII to EBCDIC conversion takes place by checking the SCLM language associated with each file/member. Then SCLM Developer Toolkit checks to see if that SCLM Language has an entry in the `TRANSLATE.conf` file with a TRANLANG keyword.

ASCII/EBCDIC language translators

SCLM Language Translator	Description
JAVA	Java source members stored as EBCDIC. Created by using sample `BWBJAVA1`.
JAVABIN	Java source members stored as ASCII. Created by using sample `BWBJAVA1`.
J2EEPART	Any J2EE files where no parsing is required and stored as EBCDIC. Created by using sample `BWBJAVAJ`.
J2EEBIN	Any J2EE files where no parsing is required and stored as binary or ASCII files. Created by using sample `BWBJAVAJ`.
TEXT	Default TEXT translator where no parsing is required and stored as EBCDIC. Created by using sample `BWBJAVAJ`.
BINARY	Default binary language translator where no parsing required. Created by using sample `BWBJAVAJ`.

Figure 22. SCLM Language Translators and ASCII/EBCDIC

Default usage is assumed to be ASCII/EBCDIC. This means that files browsed and edited in the Eclipse Plug-in may also be browsed and edited directly on host from ISPF/SCLM.

ASCII usage (binary transferred) is recommended for project migration/import and build performance, as files require no translation. This is only suitable if editing in ISPF/SCLM is not required.

Depending on the SCLM Language Translator used, source can be built in either ASCII or EBCDIC.

For cross platform useability, all deployable files such as Jar, WAR and EAR are built such that all of the contained objects are of type ASCII, regardless of whether any of the source is stored as EBCDIC.

JAVA/J2EE build note: If Java source is ASCII stored then the Build script (either language type JAVABLD or J2EEBLD) must specify the ASCII codepage using the ENCODING property variable to correctly compile the Java source.

For example:

```
<property name="ENCODING" value="ISO8859-1"/>
```

The ANT script called will use the Javac command with the ENCODING=ISO8859-1 to compile the ASCII source. The default ENCODING codepage is the EBCDIC codepage IBM-1047.

$GLOBAL member

As part of the Java/J2EE build process some additional information is required in order to successfully perform the builds. As the builds are performed in z/OS UNIX System Services, information such as the Java product location, ANT product location and the location of the SCLM Developer Toolkit code are required.

For the GUI this information is available in the HTTP server configuration file. However Java/J2EE builds are performed in the standard method used by SCLM to perform its builds, and that is to use SCLM language translators. Therefore these builds can be performed in batch or invoked directly from the ISPF interface, and in both these instances the HTTP server configuration file location is not available.

Additionally it may be required to use different versions of ANT or Java for different SCLM development groups, so to this end the $GLOBAL member can be group specific.

A sample member BWBGLOB is provided in the SBWBSAMP library. This sample member needs to be copied into SCLM type J2EEBLD in the SCLM hierarchy as member $GLOBAL and saved with a valid non parsing language, such as TEXT (as provided in language translator FLM@TEXT in the SISPMACS library).

The $GLOBAL member currently makes available the following information to the Java/J2EE build processes:

Table 5. $GLOBAL variables

Variable	Description
ANT_BIN	HFS directory path of ANT runtime **Example:** `<property name="ANT_BIN" value="/u/antdirectory/ANT/apache-ant-1.6.0/bin/ant"/>`
JAVA_BIN	HFS directory path of Java compile/runtime **Example:** `<property name="JAVA_BIN" value="/u/javadirectory/IBM/J1.3/bin"/>`
HOMEDIR	The install home directory for the SCLM Developer Toolkit WORKAREA **Example:** `<property name="HOMEDIR" value="/var/SCLMDT"/>`
CLASSPATH_JARS	z/OS HFS classpath directory used for JAVA compiles. All jars located in this directory will be used in the classpath. **Example:** `<property name="CLASSPATH_JARS" value="/var/SCLMDT/CLASSPATH"/>`
TRANTABLE	VSAM file containing the long/short name translations **Example:** `<property name="TRANTABLE" value="SCLMDT.LSTRANS.FILE"/>`

If the above variables are to be set for the all group levels in the SCLM project then it is good practice to create a single $GLOBAL member at the highest level in the hierarchy. When the Java/J2EE build translator runs it will look up the hierarchy from the group level performing the build and use the first $GLOBAL it finds in the J2EEBLD type.

Note: The $GLOBAL member must be stored as a valid saved SCLM member so this hierarchy lookup can be performed.

If different settings are required, at different development groups for example, then a $GLOBAL member can be created in each of the development groups. This may be useful if a build was required at Java 1.3 level as well as Java 1.4 level in order to test code at different Java levels.

SITE and project-specific options

A facility has been provided to allow certain settings to be made at a SITE installation level or at a specific SCLM project level. The options that can currently be configured are:

- Mandatory Change Code entry
- Deactivation of foreground Builds and Promotes
- Specification of package approval system. Currently IBM Breeze for SCLM is the supported approval system.
- Definition of Batch Build, Promote and Migrate job cards

All or none of these options can be set. If they are not set they will be defaulted in the programs. Some of these options can be set in the SITE file while others can be set at an SCLM Project specific level. Alternatively there can be no SITE specific file and options can be set at an SCLM Project level only. For job cards the user can override the job card information by using their own specified job card entered through the IDE.

This facility is activated by creating certain files in the HFS under the `/var/SCLMDT/CONFIG/PROJECT` directory, or whatever you called this directory at installation time. This directory is created at project initialization time by running job `BWBINST1`.

If you wish to set SITE specific values then you need to create a file called `SITE.conf` in the `/PROJECT` directory. A sample SITE config file is provided in the SBWBSAMP library in member BWBSITE. Copy this member to a file named `SITE.conf` in this directory and tailor the values accordingly. The following figure shows the sample SITE configuration file.

```
*
* ---------------- SITE SPECIFIC OPTIONS ------------------
*
* Below are a number of site specific options used to
* determine the behaviour of the Eclipse front-end.
* These can be overridden by creating a project specific
* options file for the SCLM project that overrides some
* or all of these options.
*
* SCM Approver processing applies to this project?
BUILDAPPROVER=NONE
PROMOTEAPPROVER=NONE
*
* Change Code entry on check-in is mandatory?
CCODE=N
*
* Foreground or On-line builds/promotes allowed for this project?
FOREGROUNDBUILD=Y
FOREGROUNDPROMOTE=Y
*
* Batch Build default jobcard
BATCHBUILD1=//SCLMBILD JOB (#ACCT),'SCLM BUILD',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
BATCHBUILD2=//*
BATCHBUILD3=//*
BATCHBUILD4=//*
*
* Batch Promote default jobcard
BATCHPROMOTE1=//SCLMPROM JOB (#ACCT),'SCLM PROMOTE',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
BATCHPROMOTE2=//*
BATCHPROMOTE3=//*
BATCHPROMOTE4=//*
*
* Batch Migrate default jobcard
BATCHMIGRATE1=//SCLMMIGR JOB (#ACCT),'SCLM MIGRATE',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
BATCHMIGRATE2=//*
BATCHMIGRATE3=//*
BATCHMIGRATE4=//*
```

Figure 23. Sample SITE specific SCLM project setting

It is also possible to have project specific configuration settings that are used to configure a single SCLM project. These will override the SITE specific values if a `SITE.conf` exists. If you wish to set Project specific values then you need to create a file called `project.conf` in the `/PROJECT` directory, where project is the SCLM Project name. A sample Project config file is provided in the SBWBSAMP library in member BWBPROJ. Copy this member to a file named `project.conf` in the `/PROJECT` directory and tailor the values accordingly. The following figure shows the sample Project configuration file.

```
*
* ---------------- PROJECT SPECIFIC OPTIONS ------------------
*
* Below are a number of project specific options used to
* determine the behaviour of the Eclipse front-end.
*
* These will override the SITE.CONF file
*
*
* SCM Approver processing applies to this project?
BUILDAPPROVER=BREEZE
PROMOTEAPPROVER=BREEZE
*
* Change Code entry on check-in is mandatory?
CCODE=Y
*
* Foreground or On-line builds/promotes allowed for this project?
FOREGROUNDBUILD=N
FOREGROUNDPROMOTE=N
*
* Batch Build default jobcard
BATCHBUILD1=//SCLMBILD JOB (#ACCT),'SCLM BUILD',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID,
BATCHBUILD2=// MSGLEVEL=(1,1)
BATCHBUILD3=//*
BATCHBUILD4=//*
*
* Batch Promote default jobcard
BATCHPROMOTE1=//SCLMPROM JOB (#ACCT),'SCLM PROMOTE',CLASS=A,MSGCLASS=X,
BATCHPROMOTE2=// MSGLEVEL=(1,1),NOTIFY=&SYSUID
BATCHPROMOTE3=//*
BATCHPROMOTE4=//*
```

Figure 24. Sample PROJECT specific SCLM project setting

Options Definition

All of the options are optional. They will be set to the default values by the product. If any of the options are specified in the `SITE.conf` or the `project.conf` then they will be used.

Table 6. SITE/Project options

`BUILDAPPROVER=approval product/NONE`	Specify the name of the approval product used for the build process. Currently the only supported product is Breeze for SCLM. Default is NONE.
`PROMOTEAPPROVER=approval product/NONE`	Specify the name of the approval product used for the promote process. Currently the only supported product is Breeze for SCLM. If the PROMOTEAPPROVER is set to BREEZE then the Breeze specific fields will be displayed during a promote. Default is NONE.
`CCODE=N/Y`	Specify Y to make change code entry on check-in a mandatory field. Default is N such that Change Code entry is not mandatory.
`FOREGROUNDBUILD=Y/N`	Specify N to restrict foreground builds. Default is Y such that foreground builds are allowed.

Table 6. SITE/Project options (continued)

FOREGROUNDPROMOTE=Y/N	Specify N to restrict foreground promotes. Default is Y such that foreground promotes are allowed.
BATCHBUILD1=Job card 1 BATCHBUILD2=Job Card 2 BATCHBUILD3=Job Card 3 BATCHBUILD4=Job Card 4	Set a default batch job card for the build process. Different projects may use different account codes or Job class so the option of specifying project specific job cards allows for this scenario.
BATCHPROMOTE1=Job card 1 BATCHPROMOTE1=Job card 2 BATCHPROMOTE1=Job card 3 BATCHPROMOTE1=Job card 4	Set a default batch job for the Promote process. Different projects may use different account codes or Job class so the option of specifying project specific job cards allows for this scenario.
BATCHMIGRATE1=Job card 1 BATCHMIGRATE1=Job card 2 BATCHMIGRATE1=Job card 3 BATCHMIGRATE1=Job card 4	Set a default batch job for the Migrate process. Different projects may use different account codes or Job class so the option of specifying project specific job cards allows for this scenario.

Chapter 3. CRON-initiated Builds and Promotes

Though most Builds and Promotes will be initiated via the Developer Toolkit client, there is a provision to set up Build and Promote configuration files within the z/OS HFS and to initiate these builds or promotes via the CRON (time) service within UNIX System Services. Using this method the DT Client is not required, as the relevant Build and Promote parameters are read from a HFS configuration file and passed to the DT Host component for SCLM processing.

Below is a description of the SCLM Developer Toolkit samples that provide CRON initiated Builds and Promotes. These samples are available in the installed Developer Toolkit SBWBSAMP dataset.

Sample member	Description
BWBCRON1	This REXX sample calls the SCLM Developer Toolkit host interface and passes the function parameters. Output from the function process by default is displayed to STDOUT but may be re-directed to a HFS file or log. This sample may be copied into the HFS into a directory path of the users choice to run. The sample will need to be customized as detailed within the sample. This REXX sample must be run in conjunction with input from sample BWBCRONB for a build or sample BWBCRONP for a promote.
BWBCRONB	This REXX sample sets up the Build parameter input string which will be passed to module BWBCRON1. The sample requires user customization to update all required build parameters. This sample must be copied into a user determined HFS directory (optionally renamed) to be run with sample BWBCRON1.
BWBCRONP	This REXX sample sets up the Promote parameter input string which will be passed to module BWBCRON1. The sample requires user customization to update all required promote parameters. This sample must be copied into a user determined HFS directory (optionally renamed) to be run with sample BWBCRON1.

Figure 25. Sample CRON members

STEPLIB and PATH requirements

The PATH and STEPLIB variables in either the system wide profile (`/etc/profile`) or the users profile (`/u/userid/.profile`) will need to be set to locate the cron jobs ($PATH) and locate the SCLM Developer Toolkit modules ($STEPLIB) if the DT dataset is not in the LINKLIST.

Example:

The samples BWBCRON1 and BWBCRONB are copied to a test directory /var/SCLMDT/CRONJOBS. The following HFS PATH and STEPLIB variables are set in /etc/profile:

```
PATH=/var/SCLMDT/CRONJOBS:$PATH
STEPLIB=BWB.V1R1M0.SBWBLOAD:$STEPLIB
```

CRON Build job execution

Once the CRON jobs are added to the PATH variable they can be run by piping the output from the parameter_exec into the processing_exec. The output can then be directed to an output log file.

Syntax

```
parameter_exec | processing_exec > output.log
```

The "|" is the z/OS UNIX System Services pipe symbol.

Invocation Example:

Using the sample names as provided the CRON build exec could be invoked as follows:

```
BWBCRONB | BWBCRON1 >bwbcronb.log
```

Additionally the above sample Build execution could be added to a CRONTAB file to run at 7.30pm Monday- Friday:

```
30 19 * * 1-5 BWBCRONB|BWBCRON1 >bwbcronb.log ;
```

For further information on the CRON services available and the CRONTAB format refer to the following manuals:

- z/OS UNIX System Services Commands
- z/OS UNIX System Services Planning

Alternatively use the online manual help (man) under z/OS UNIX System Services:

- man cron
- man crontab
- man at

CRON Build job samples

Below are the BWBCRON1 and BWBCRONB job samples as provided in the SBWBSAMP library.

```
/* REXX */
/* Customize STEPLIB and CGI_VCMPATH BELOW */
/*
The STEPLIB should reflect the install load library for SCLM Developer
toolkit.
If this dataset resides in the LINKLIST then set STEPLIB to '' .
*/
STEPLIB = 'BWB.V1R1M0.SBWBLOAD'
/*
The Environment variable CGI_VCMPATH determines the HOME directory
path where the configuration files reside for SCLM Developer Toolkit.
This was determined by the install directory specified in install
job BWBINST1. By default /var/SCLMDT .
*/
CGI_VCMPATH = '/var/SCLMDT'
/* */
/* SAMPLE USEAGE */
/*
COMMAND : BWBCRONB|BWBCRON1 >BWBCRONB.log
(passes build parameter list to BWBCRON1 & outputs to BWBCRONB.log)
*/
/* DO NOT ALTER BELOW */
CALL ENVIRONMENT 'STEPLIB',STEPLIB
CALL ENVIRONMENT 'CGI_VCMPATH',CGI_VCMPATH
CALL BWBINT
EXIT
```

Figure 26. Sample CRON Build Exec

```
/* REXX */
/* SAMPLE BUILD PARAMETER FILE USED FOR CRON INITIATED BUILDS */
/* Update Build parameters below */
/* if parameter required as Blank then set as '' */
FUNCTION = 'BUILD'
PROJECT  = 'PROJ1'   /* SCLM Project            */
PROJDEF  = ''        /* Alt proj definition     */
TYPE     = 'SOURCE'  /* SCLM Type               */
MEMBER   = 'TESTMEM' /* SCLM Member name        */
GROUP    = 'DEV1'    /* SCLM Group              */
GROUPBLD = ''        /* Build at Group          */
REPDGRP  = 'DEV1'    /* Users Development group */
BLDREPT  = 'Y'       /* Generate Build report   */
BLDLIST  = 'Y'       /* Generate List on error  */
BLDMSG   = 'Y'       /* Generate Build Messages */
BLDSCOPE = 'N'       /* Build Scope E/L/N/S     */
BLDMODE  = 'C'       /* Build Mode C/F/R/U      */
BLDMSGDS = ''        /* Message dataset         */
BLDRPTDS = ''        /* Report dataset          */
BLDLSTDS = ''        /* list dataset            */
BLDEXTDS = ''        /* Exit dataset            */
SUBMIT   = 'BATCH'   /* Online or Batch         */
/* DO NOT ALTER PARM BUILD VARIABLE BELOW */
PARM1 = 'SCLMFUNC='FUNCTION'&PROJECT='PROJECT'&PROJDEF='PROJDEF||,
'&TYPE='TYPE'&MEMBER='MEMBER'&GROUP='GROUP'&GROUPBLD='GROUPBLD||,
'&REPDGRP='REPDGRP'&BLDREPT='BLDREPT'&BLDLIST='BLDLIST||,
'&BLDMSG='BLDMSG'&BLDSCOPE='BLDSCOPE'&BLDMODE='BLDMODE||,
'&BLDMSGDS='BLDMSGDS'&BLDRPTDS='BLDRPTDS'&BLDLSTDS='BLDLSTDS||,
'&BLDEXTDS='BLDEXTDS'&SUBMIT='SUBMIT
/* outputs parameter string as input to BWBCRON1 */
SAY PARM1
```

Figure 27. Sample Build parameter file

Appendix A. Long/short name translation table

Currently core SCLM does not support the use of code storage with file (member) names greater than eight characters.

Code such as Java and other PC client code inherently have much greater name lengths and even incorporate path information (packaging) as part of the name. This causes the need for code with named parts greater than eight characters to go through a long/short name conversion utility to enable these parts to be stored within SCLM with an eight character (or less) name length.

A longname to shortname translation table stores the matching longnames (real name) against the short names (as stored in SCLM). These tables are controlled and accessed by SCLM and reside in a VSAM dataset. This functionality has been introduced into SCLM with the PTF that addresses APAR OA11426 for z/OS V1.4 and later.

The conversion algorithm follows these steps:

1. The translate prefix consists of the first two characters (uppercase) of the long program/file name (that is, the last file name after "/" character in multi-packaging format). If the first two characters are not valid as a prefix for a host member name (because they contain invalid special characters) then the prefix is "XX". Special cases, such as a single character alphabetic name (/u/test/A or /u/test/A.java), are also assigned the prefix of "XX".
2. The last six characters are numerically assigned the next numeric sequential number available in the translate table.

For example:

Longname	Shortname in SCLM PDS or PDSE
com/ibm/workbench/testprogram.java	TE000001
source/plugins/Phantom/.classpath	XX000001

Technical summary of the SCLM Translate program

SCLM program FLMLSTRN was created to read and update the VSAM translation table. SCLM Developer Toolkit uses this program to read and update correlating longnames and shortnames.

The VSAM file used to store the translation table is a variable length KSDS with an alternative index and path defined. A sample job is provided in SCLM to allocate this VSAM file.

The internal structure of the VSAM cluster is:

```
1 ls_record,
  3 ls_short_name  char(08),
  3 ls_lngname_key char(50),
  3 ls_long_name   char(1024);
```

Sample JCL to allocate the Long/Short name translation VSAM file can be seen in "Step 5: Configure long/short name table VSAM file" on page 7.

Note: The following technical information on SCLM Translate table function calls is supplied as information only and is not required to enable any SCLM Developer Toolkit functionality.

The program FLMLSTRN is invoked via the ISPF SELECT service passing one of the parameters listed in Table 7.

Syntax

```
"SELECT PGM(FLMLSTRN) PARM(keyword)"
```

Invocation example:

```
"SELECT PGM(FLMLSTRN) PARM(TRANSLATE)"
```

Table 7. Long/Short name translation parameters

Keyword Record	Processing	Description
FINDLONG	Single	Find a long name for a given short name
FINDSHORT	Single	Find a short name for a given longname
TRANSLATE	Single	Find a short name if it exists or allocate a new short name if it does not exist
MIGRATE	Multiple	Search for multiple longnames
IMPORT	Multiple	Search for multiple shortnames

Single long/short name record processing

FINDLONG Processing

- The VSAM cluster allocated to DD LSTRANS is opened in read mode.
- The short name is retrieved from the ISPF variable FLMLSSHR and this short name is used to read the VSAM file.
- If the record is not found a message is returned via the ISPF variable FLMLSERR stating long name was not found.
- If the long name was found it is returned in the ISPF variable FLMLSLNG.
- The VSAM cluster is closed.

FINDSHORT Processing

- The VSAM Path allocated to DD LSTRNPTH is opened in read mode.
- The long name is retrieved from the ISPF variable FLMLSLNG.
- The last 50 bytes of the long name is used to read the path.
- If a record is not returned a message is returned via the ISPF variable FLMLSERR stating short name was not found.
- If a record is returned the long name in the VSAM record is checked against the long name in the ISPF variable FLMLSLNG.
- If it doesn't match the VSAM records are read and compared until the ls_lngname_key doesn't match or the long name is found.

  Note: The `ls_lngname_key` allows duplicates as it is possible to have a VSAM record with the same ls_lngname_key but different long name.
- If the short name was found it is returned in the ISPF variable FLMLSSHR.
- The VSAM path is closed.

TRANSLATE Processing

The processing is the same as for FINDSHORT.

- If the short name is found no further processing is performed.
- If the short name is not found the VSAM cluster allocated to DD LSTRANS is opened in update mode.
- The file name is determined by finding the last '/' or '\' in the long name.
- The first 2 bytes of the file name are used to look up the VSAM file prefix record which contains a number.
- The file prefix and number will be used to generate the short name (for example, PR000123).
- The short name generated (PR000123) is used to check VSAM file to determine if the short name is being used.
- If it is the prefix number is incremented and short name again checked.
- This process continues until we find a short name that is not being used.
- The prefix record is updated and then the new translate record is added.
- The short name is returned in the ISPF variable FLMLSSHR.
- The VSAM cluster is closed.

Multiple long/short name record processing

MIGRATE and IMPORT are functions that were introduced to improve performance with large numbers of longnames being translated (MIGRATE) or large numbers of shortnames being searched for (IMPORT).

Both functions, "MIGRATE" and "IMPORT", read a variable blocked sequential file with LRECL=1036 which is allocated as DD LSTRNPRC.

Before invocation this file will contain the shortnames or longnames depending on the function called and in the correct format and column.

After invocation LSTRNPRC will contain both the shortname and correlating longname.

The format of the file is:

```
1 pr_record,
  3 pr_short_name  char(08),
  3 pr_long_name   char(1024);
```

IMPORT processing

- The VSAM cluster allocated to DD LSTRANS is opened in read mode and the processing file allocated to DD LSTRNPRC is opened for update.
- For each of the records on the processing file, the short name is used to read the VSAM translation file. If a record is found the processing file is updated with the long name.
- The VSAM cluster/processing files are closed.

MIGRATE processing

- The VSAM cluster allocated to DD LSTRANS is opened in read mode and the processing file allocated to DD LSTRNPRC is opened for update.
- For each of the records on the processing file, the long name is used to read the VSAM file. If a record is found the processing file record is updated with its

corresponding short name otherwise LSTRANS is opened in update mode to add new long name/short name entries and the new shortname generated is written back to the LSTRNPRC file.

- The VSAM cluster/processing files are closed.

Here is some sample REXX to invoke the long/short name translation process:

```
/* REXX ***************************************************************/
/* Sample to translate longname to a shortname */
/**********************************************************************/
   Address TSO
   "FREE FI(LSTRANS)"
   "FREE FI(LSTRNPTH)"
   "ALLOC DD(LSTRANS) DA('SCLMDT.LSTRANS.FILE') SHR REUSE"
   "ALLOC DD(LSTRNPTH) DA('SCLMDT.LSTRANS.FILE.PATH') SHR REUSE"
   /* Create shortname for longname com/ibm/phantom.txt */
   FLMLSLNG = "com/ibm/phantom.txt"
   Address ISPEXEC "VPUT (FLMLSLNG) PROFILE"
   Address ISPEXEC "SELECT PGM(FLMLSTRN) PARM(TRANSLATE)"
   LSRC=RC
   If LSRC > 0 Then
   Do
      Address ISPEXEC "VGET (FLMLSERR,FLMLSER1) PROFILE"
      Say "LS ERROR LINE 1 ==>" FLMLSERR
      Say "LS ERROR LINE 2 ==>" FLMLSER1
      Return
   End
   Else
   Do
      Address ISPEXEC "VGET (FLMLSSHR,FLMLSLNG) PROFILE"
      Say " Shortname = " FLMLSSHR
      Say " Longname = " FLMLSLNG
   End
   Address TSO
   "FREE FI(LSTRANS)"
   "FREE FI(LSTRNPTH)"
```

Figure 28. Sample REXX for Translate module invocation

Appendix B. Messages and codes

BWB00001 ERROR in Allocation/Write to : *data set name*

Explanation: An attempt to write the client parameter input to a temporary dataset has failed.

User Response: Review the operations log to further determine any errors.

BWB00002 Processing terminates

Explanation: The requested function is cancelled and the ISPF/SCLM session is ended.

User Response: Review additional error messages and the operations log to determine error.

BWB00003 Following function has failed : *function*

Explanation: Indicated the failing Function or operation request.

User Response: Review additional error messages and the operations log to determine error.

BWB00004 ISPF initialization has failed

Explanation: The ISPF/SCLM session has failed to initialize. Processing on that function request is terminated.

User Response: Review additional error messages and the operations log to determine error.

BWB00005 Review your ISPF properties and Loadlib

Explanation: Additional error messaging to indicate a reason for ISPF/SCLM initialization failure.

User Response: The Systems programmer should review the ISPF.conf configuration file in the HFS CONFIG directory to determine the correct ISPF dataset allocations have been set.

BWB00006 Ensure the Loadlib data set is APF authorized

Explanation: The SCLM Developer Toolkit Loadlib dataset may not be APF authorized resulting in ISPF/SCLM initialization failure.

User Response: The Systems programmer should ensure the Loadlib is APF authorized or that no non-APF authorized datasets have been included in the HTTP server STEPLIB.

BWB00007 ISPF or SCLM service has abended

Explanation: An ISPF/SCLM internal error has caused the session to terminate. The function request is cancelled.

User Response: Review the operations log to further determine any errors.

BWB00008 Review Log for error details

Explanation: Review the Client operations log for further messages or error analysis.

User Response: To review the operations log, right click and select TEAM then operations log.

BWB00009 Error reading ISPF.conf

Explanation: The configuration file ISPF.conf located in the z/OS HFS at installdirectory/ISPF.conf could not be read

User Response: The Systems Programmer should ensure the configuration file exists and the user has read access.

BWB00010 Ensure file exists in directory specified by environment variable CGI_VCMPATH set in the server configuration

Explanation: Configuration files such as ISPF.conf should be located in the directory specified by CGI_VCMPATH

User Response: The Systems Programmer should check CGI_VCMPATH in httpd.env.

BWB00011 If error on ISPSPROF table then a background session may already be using the profile data set : data set name

Explanation: Two concurrent sessions have been initiated and using the same ISPF profile dataset.

User Response: Check that another session is not running concurrently and preventing this session to establish.

BWB00012 Error in ISPF data set allocation : See error message below

Explanation: An error in one of the ISPF dataset allocations. Specific allocation error messages follow.

User Response: The Systems Programmer should

check ISPF.conf to ensure valid ISPF DD names and correct ISPF datasets.

BWB00013 Error in allocating the following DD and data set names : *error message*

Explanation: The following dataset could not be allocated to that DD name.

User Response: The Systems Programmer should review the DD and dataset allocation (ISPF.conf).

BWB00014 Check ISPF configuration file ISPF.conf on the host is correct

Explanation: An error in the ISPF dataset allocations has occurred.

User Response: The Systems Programmer should check ISPF.conf to ensure valid ISPF DD names and correct ISPF datasets.

BWB00015 * Edit cancelled *****

Explanation: EDIT processing has been cancelled due to an error.

User Response: Review additional error messages and the operations log to determine error.

BWB00016 Selected group not in development group hierarchy

Explanation: The group selected is not valid as it does not reside in the USERS development hierarchy.

User Response: Choose a valid group or operate from a different development group.

BWB00017 Unable to locate data set for : *project group type*

Explanation: An error has occurred in locating a valid dataset for that particular project/group/type.

User Response: The SCLM administrator should check that a valid dataset exists for that project/group/type.

BWB00018 * Edit/Browse cancelled *****

Explanation: An error has occurred in EDIT/BROWSE and the function is cancelled.

User Response: Review additional error messages and the operations log to determine error.

BWB00019 Member selected resides in lower hierarchy group

Explanation: Selected member is not valid for operation as a valid member resides in a lower hierarchy group.

User Response: Select member from lower hierarchy group.

BWB00020 Member should be selected for from group : *group*

Explanation: Additional messaging indicating which member should be selected for this operation.

User Response: Select recommended member and repeat function.

BWB00021 Repository view mismatch

Explanation: The USERS repository view and SCLM are not synchronized.

User Response: Re-populate project to synchronize the view with SCLM.

BWB00022 Re-filter using Populate SCLM Project View or re-synchronize IDE view

Explanation: The repository view or IDE view are not synchronized with SCLM.

User Response: Re-populate project if explorer mode or developer mode are not synchronized. Otherwise re-synchronize IDE view.

BWB00023 * Add new member function cancelled *****

Explanation: The Add new member function has been cancelled due to errors.

User Response: Review additional error messages and the operations log to determine error.

BWB00024 Member does not reside in user's development group

Explanation: An invalid SCLM operation has occurred as the member does not reside in the user's development group.

User Response: Select a valid member.

BWB00025 * No member to Browse *****

Explanation: A valid SCLM member with text was not found.

User Response: None

BWB00026 * Member copy into HFS failed *****

Explanation: The operation involves copying the member to the z/OS HFS. This operation failed.

User Response: The Systems programmer should review the operations log for further explanation of this error.

BWB00027 Edit failed - copy failure : *data set name(member)* **to** *data set name*

Explanation: As part of the EDIT function a copy of the SCLM member to a temporary dataset has failed.

User Response: Review additional error messages and the operations log for further information.

BWB00028 Error returned on member lock

Explanation: SCLM lock on member for edit failed. Member may already be locked for edit by another user.

User Response: Issue an SCLM status to determine if member is locked for EDIT by another user.

BWB00029 Binary formatted members not valid for Edit

Explanation: Binary object type members of NON-EDIT are not eligible to be selected for EDIT. Function cancelled.

User Response: None

BWB00030 * Copy to HFS failed *****

Explanation: The operation involves copying the member to the z/OS HFS. This operation failed.

User Response: The Systems programmer should review the operations log for further explanation of this error.

BWB00031 * Error in ASCII/EBCDIC translation *****

Explanation: An error has occurred in the translation from ASCII to EBCDIC when checking in a file.

User Response: Ensure the file to be checked in is not empty or contains invalid special characters. Otherwise the ASCII EBCDIC codepages used should be verified.

BWB00032 * Error in member copy to SCLM *****

Explanation: The member being checked in wasn't successfully copied into the MVS™ dataset. The operation has failed.

User Response: The Systems programmer should review the operations log for further explanation of this error.

BWB00033 * Save function failed *****

Explanation: The SAVE function has failed.

User Response: Review additional error messages and the operations log to determine error.

BWB00034 * Save function cancelled *****

Explanation: The check-in function to SCLM was cancelled.

User Response: Review additional error messages and the operations log to determine reason.

BWB00035 Member not previously locked for Edit

Explanation: Associated with the SAVE / CHECK-IN function where a member has to be SCLM locked or checked out first.

User Response: The USER should lock or check-out the member first using the EDIT function.

BWB00036 Error returned from ACCTINFO status on member

Explanation: The requested function internally performs an SCLM ACCTINFO request which has failed.

User Response: Review additional error messages and the operations log to determine error.

Check the validity of the member.

BWB00037 * Save function successful *****

Explanation: The SAVE or Check-in operation was successful.

User Response: Smile

BWB00038 Error returned on SAVE for :

MEMBER = *member*
PROJECT = *project*
GROUP = *group*
TYPE = *type*

User Response: Lists member details for the failed SAVE request.

User Response: Review additional error messages and the operations log to determine error.

BWB00039 Delete successful for member : *member*

Explanation: The selected member was deleted successfully.

User Response: None

BWB00040 Delete failed for member : *member*

Explanation: Delete function failed for requested member.

User Response: Review additional error messages and the operations log to determine error.

BWB00041 Error returned on delete

Explanation: The SCLM delete command has returned a non-zero return code.

User Response: Review additional error messages and the operations log to determine error.

BWB00042 Please check log file for information messages

Explanation: Additional messaging may be contained in the operations log.

User Response: To review the operations log, right click and select TEAM then operations log.

BWB00043 Unlock failed for member : *member*

Explanation: The SCLM unlock command on a cancel EDIT request has failed.

User Response: Review the SCLM status of the member.

BWB00044 Unlock successful for member : *member*

Explanation: A member previously locked for EDIT has been successfully unlocked.

User Response: None

BWB00045 Failed allocation of build parms DSN : *data set name*

Explanation: The temporary dataset USERID.TEMP.VCMISPF containing the build parameters could not be allocated as DISP=SHR. Processing terminates.

User Response: The Systems programmer should review the operations log for additional error messages.

BWB00046 Failed read of build parms

Explanation: The temporary dataset containing the build parameters was allocated but an error occurred while reading or the dataset was empty.

User Response: The Systems programmer should review the operations log for further explanation of this error.

BWB00047 Warning - Unable to allocate build messages data set

Explanation: The specified Build message dataset could not be allocated. Processing continues.

User Response: Review the Build messages dataset that was user specified from the Build request. Review MVS SYSLOG for any security violations.

BWB00048 Warning - Unable to allocate build report data set

Explanation: The specified Build report dataset could not be allocated. Processing continues.

User Response: Review the Build report dataset that was user specified from the Build request. Review MVS SYSLOG for any security violations

BWB00049 Warning - Unable to allocate build list data set

Explanation: The specified Build list dataset could not be allocated. Processing continues.

User Response: Review the Build list dataset that was user specified from the Build request. Review MVS SYSLOG for any security violations

BWB00050 Warning - Unable to allocate build exit data set

Explanation: The specified Build exit dataset could not be allocated. Processing continues.

User Response: Review the Build exit dataset that was user specified from the Build request. Review MVS SYSLOG for any security violations

BWB00051 Build failed

Explanation: The build request has failed

User Response: Review additional error messages and the operations log to determine error.

BWB00052 Error reading build messages

Explanation: The Build messages dataset could not be read.

User Response: Review operations log for successful dataset allocation.

BWB00053 Error reading build report

Explanation: The Build report dataset could not be read.

User Response: Review operations log for successful dataset allocation.

BWB00054 Error reading build listing

Explanation: The Build listing dataset could not be read.

User Response: Review operations log for successful dataset allocation.

BWB00055 Error reading J2EEBLD report

Explanation: The Build J2EEBLD report dataset could not be read.

User Response: Review operations log for successful dataset allocation.

BWB00056 BATCH skeleton error (Administrator customization error)

Explanation: The SCLM batch skeletons FLMDSU$ FLMB$ could not be included. Build processing terminates.

User Response: Review MSG BWB00057

BWB00057 Check SCLM Developer Toolkit Skeleton concatenated in ISPSLIB as specified in file ISPF.conf

Explanation: Extended messaging from MSG BWB00056.

User Response: The Systems programmer should check that the base SCLM skeletons are allocated to ISPSLIB in the HFS configuration file ISPF.conf

BWB00058 No BATCH submission information returned

Explanation: No BATCH Job information was returned or captured from TSO.

User Response: Review the MVS SYSLOG to determine if a batch job was submitted successfully or the Job resides in an SDSF queue

BWB00059 * Promote cancelled *****

Explanation: The Promote function has been cancelled.

User Response: Review MSG BWB00016

BWB00060 Warning - Unable to allocate messages data set

Explanation: The temporary dataset to receive SCLM messages USERID.VCM.TEMP.FLMMSG could not be allocated. Processing continues.

User Response: The Systems programmer should review the operations or MVS log for additional error messages.

BWB00061 Failed allocation of promote parms DSN : *data set name*

Explanation: The temporary dataset USERID.TEMP.VCMISPF containing the build parameters could not be allocated as DISP=SHR. Processing terminates.

User Response: The Systems programmer should review the operations log for additional error messages.

BWB00062 Failed read of promote parms

Explanation: The temporary dataset containing the build parameters was allocated but an error occurred while reading or the dataset was empty.

User Response: The Systems programmer should review the operations log for further explanation of this error.

BWB00063 Warning - Unable to allocate promote messages data set

Explanation: The specified Promote message dataset could not be allocated. Processing continues.

User Response: Review the Promote messages dataset that was user specified from the promote request. Review MVS SYSLOG for any security violations.

BWB00064 Warning - Unable to allocate promote report data set

Explanation: The specified Promote report dataset could not be allocated. Processing continues.

User Response: Review the Promote report dataset that was user specified from the promote request. Review MVS SYSLOG for any security violations

BWB00065 Warning - Unable to allocate promote exit data set

Explanation: The specified Promote exit dataset could not be allocated. Processing continues.

User Response: Review the Promote exit dataset that was user specified from the promote request. Review MVS SYSLOG for any security violations

BWB00066 Warning - Unable to allocate Authcode messages data set

Explanation: The temporary authcode messages dataset USERID.VCM.TEMP.AUTHMSG could not be allocated. Processing continues.

User Response: Review MVS SYSLOG for any security violations

BWB00067 Ensure build was successful

Explanation: Promote failed. Possible reason is invalid Build map.

User Response: Review previous build request or view build map.

BWB00068 Breeze is configured for this project, check log for Breeze messages

Explanation: Informational message that the Breeze flag is set and to review additional Breeze messages.

User Response: None

BWB00069 Error reading promote messages

Explanation: The Promote messages dataset could not be read.

User Response: Review operations log for successful dataset allocation.

BWB00070 Error reading promote report

Explanation: The Promote report dataset could not be read.

User Response: Review operations log for successful dataset allocation.

BWB00071 Populate view of project failed

Explanation: Either Project Initialization has failed or the development group specified is invalid for the project.

User Response: Review any additional SCLM messages in the operation log and check your development group, Project and filter parameters.

BWB00072 Reading ARCHDEF directly as no BLDMAP found

Explanation: Populate project by archdef will generate the request using DBUTIL and the build map for the archdef. If no build map exists (i.e. the archdef has not been built) then members will be read directly from the archdef.

User Response: None

BWB00073 Only members in lowest hierarchy (latest) listed

Explanation: Additional to MSG BWB00072. As there is no archdef build map from which to generate the report, the populate will only report on members at the lowest group in the hierarchy.

User Response: If not acceptable then use a populate project filter other than archdef or build the archdef first to populate full list.

BWB00074 No Project view information returned

Explanation: No project information was returned using the specified filters.

User Response: Review your filter parameters.

BWB00075 Review your filter parameters

Explanation: Information to MSG BWB00074.

User Response: None

BWB00076 The following ARCHDEF include was not found in the hierarchy : *include name*

Explanation: An INCLD statement for a member was found in the selected archdef but the member was not found in any group up the development hierarchy.

User Response: If old or invalid member then remove from archdef.

BWB00077 * Update function failed *****

Explanation: The update function failed as the member could not be locked for edit/update.

User Response: Review SCLM status of member.

BWB00078 Error returned on LOCK

Explanation: Unable to obtain lock on member for EDIT or Update.

Member may already be locked (checked out) by another user.

User Response: Review SCLM status of member.

BWB00079 * Update function cancelled *****

Explanation: Update is cancelled as member was not previously locked for edit.

User Response: Review SCLM status of member.

BWB00080 Member not previously locked for edit

Explanation: informational

User Response: Retry function.

BWB00081 * Update function successful *****

Explanation: informational

User Response: None

BWB00082 * AUTHCODE retrieval failed *****

Explanation: Unable to retrieve authcode on selected member.

User Response: Review SCLM status of member.

Review any SCLM messages in operation log.

BWB00083 * AUTHCODE update failed *****

Explanation: The update of authcode on the member has failed. Processing terminates.

User Response: Review SCLM status of member.

Review any SCLM messages in operations log.

BWB00084 * VERINFO command failed *****

Explanation: The VERINFO request failed with a return code > 8.

User Response: Review any SCLM messages in operations log.

BWB00085 No Version records were found matching criteria

Explanation: Informational

User Response: None

BWB00086 * VERRECOV command failed *****

Explanation: A version recover request has failed with a return code > 0.

User Response: Review any SCLM messages in operations log.

BWB00087 * VERDEL command failed *****

Explanation: A version delete request has failed with return code > 0.

User Response: Review any SCLM messages in operations log.

BWB00088 Severe error in catalog search routine

Explanation: An error has occurred in the IGGCSI00 Catalog search program with a project list request.

User Response: The Systems programmer should review the operations and MVS log for further error messages. Retry request.

BWB00089 Warning - No Projects returned on filter : *project filter*

Explanation: Informational.

User Response: Review project filter request.

BWB00090 Error reading translate table : *file name*

Explanation: An error has occurred reading the translate table. Processing continues but there will be errors in the longname to shortname retrieval and updates.

User Response: The Systems Programmer should verify correct translate table name: CGI_TRANTABLE variable in the httpd.env configuration file.

Run the install IVP to also verify installation.

BWB00091 Long/short name translation may be in error

Explanation: A previous error with the Longname to shortname translation file has occurred.

User Response: The Systems programmer should review the operations log for further error messages.

BWB00092 Error reading file : *file name*

Explanation: The displayed HFS file contained the files to migrate could not be read. Migration terminates.

User Response: Review the operations log for further error messages.

BWB00093 Error returned from SCLMINFO file allocation

Explanation: Error on allocation of new temporary dataset userid.vcm.temp.projinf

DCB: SPACE(1,1,20) TRKS RECFM(FB) LRECL(40) BLKSIZE(27960) VIO

User Response: Review the MVS SYSLOG for additional messages for dataset allocation failure.

BWB00094 Error returned from SCLMLOAD file allocation

Explanation: Unable to allocate DISP=SHR the dataset Project.projdefs.load.

User Response: Determine the dataset Project.projdefs.load exists, where project is the project name specified.

BWB00095 Error returned on call to ISRFLMGI

Explanation: Unable to call ISPF/SCLM module ISRFLMGI for retrieving group information for project.

User Response: None

BWB00096 Unable to re-allocate SCLMINFO (no groups returned)

Explanation: Unable to allocate temporary dataset for project group information retrieval.

User Response: Review the operations log for further error messages.

BWB00097 Unable to read SCLMINFO (no groups returned)

Explanation: Unable to read allocated SCLMINFO temporary dataset. No project information will be returned.

User Response: Review the operations log for further error messages.

BWB00098 Group list has failed

Explanation: No group information for selected project is returned due to function failure.

User Response: Review the operations log for further error messages.

BWB00099 Build map deleted for member : *member*

Explanation: Informational.

Delete request has deleted build map.

User Response: None

BWB00100 Jobid must have valid Jobname in request

Explanation: No Jobname was passed through in JOB status request with JOBID.

User Response: Retry with valid JOB Name specified.

BWB00101 No Job details passed - Defaults to USERID

Explanation: No Jobname or Jobid details were passed. Function will report on all jobs starting with USERID as Jobname.

User Response: None

BWB00102 For job retrieval a valid Jobname and Jobid must be provided

Explanation: No valid Jobname / Jobid was passed for Job output retrieval.

User Response: Retry with valid Jobname/Jobid.

BWB00103 No job output found

Explanation: The requested Batch job was not found on the SDSF queue.

User Response: Check on z/OS in SDSF if the batch job still resides on an output queue or has been purged.

BWB00104 Warning - Refer to additional SCLM messages

Explanation: Additional SCLM warning messages have been issued.

User Response: View additional SCLM messages in operations log.

BWB00105 Error - SCLM is unable to load the SCLM table

Explanation: Error on SCLMINFO call for project/Project definition.

User Response: Check Project and project definition are valid.

BWB00106 Check that the following projdef is valid : *Project*

Explanation: Informational.

Extends MSG BWB00105.

User Response: Check Project and project definition are valid.

BWB00107 Error - The maximum application ID limit was exceeded

Explanation: SCLMINFO request has returned return code 12.

User Response: Check Project and project definition are valid. Inform SCLM administrator of error.

BWB00108 Errors initializing project : *Project*

Explanation: Extends MSG BWB00107

User Response: Reference MSG BWB00107.

BWB00109 Error - An invalid version of the SCLM table was loaded

Explanation: SCLMINFO request has returned return code 16.

User Response: Check Project and project definition are valid. Inform SCLM administrator of error.

BWB00110 Error - An invalid parameter list was passed

Explanation: SCLMINFO request has returned return code 32.

User Response: Check Project and project definition are valid. Inform SCLM administrator of error.

BWB00111 Error - Refer to SCLM messages manual on SCLMINFO codes

Explanation: The SCLM messages manual will give additional information on the SCLMINFO return codes.

User Response: Review the SCLM messages manual.

BWB00112 No project specific entries in project config file : *file name*

Explanation: Informational: The SITE.conf file has not been tailored for use.

User Response: None

BWB00113 Failed allocation of migrate parms DSN : *data set name*

Explanation: The Temporary MIGRATE dataset USERID.TEMP.MIGRATE.VCMISPF could not be allocated DISP=SHR. Processing terminates.

User Response: The Systems programmer should review the operations log for additional error messages.

BWB00114 Failed read of migrate parms

Explanation: Error reading allocated parameter dataset.

User Response: The Systems programmer should review the operations log for additional error messages

BWB00115 Error on directory : *directory name*

Explanation: Error in creating or locating specified directory in HFS which is used to process the current function.

User Response: The Systems programmer should review the operations log for additional error messages

BWB00116 PUT command failed for the following : *member info*

Explanation: A copy of a member from an SCLM dataset into a HFS directory has failed.

User Response: Check SCLM status of member.

Review the operations log for additional error messages

BWB00117 Error threshold of 5 errors reached in copy

Explanation: An error threshold of 5 bad member copies between SCLM and the HFS directory has occurred in the Java/J2ee build translators. Processing terminates.

User Response: Review the operations log for additional error messages.

Rectify problem and retry function request.

BWB00118 Errors in code page conversions - Review LOG

Explanation: Errors have occurred in the ASCII / EBCDIC codepage conversions.

User Response: Review the operations log for additional error messages.

BWB00119 JAR Failed on following file : *file name*

Explanation: An error has occurred in creating a java JAR file. This function uses java to JAR up a zip file for transferring files across the network.

User Response: Review the operations log for additional error messages.

The Systems Programmer should ensure a valid java path exists in httpd.env and check for space problems within the HFS directory WORKAREA.

BWB00120 JAVA PATH in httpd.env file may not be valid. Check with your Administrator

Explanation: An error has occurred is issuing a JAVA command.

User Response: The Systems Programmer should ensure a valid java path exists in httpd.env for the HTTP server.

BWB00121 Error found in DBUTIL report - Review LOG

Explanation: A failure has occurred with an SCLM DBUTIL request for producing a member list. Further error messages may be displayed.

User Response: Review the operations log for additional error messages.

BWB00122 No project information returned by filter

Explanation: The USER specified filter list has returned no project information.

User Response: Review your filter parameters and retry.

BWB00123 SCLM import successful for the following number of members : *number of members*

Explanation: Informational

User Response: None

BWB00124 **No files to migrate specified in migrate file :** *file name*

Explanation: No files were found to be migrated.

User Response: If an error review the operations log for additional error messages.

BWB00125 **Invalid Host name convention for :** *long name*

Explanation: The file to be stored is not defined as a long language type but is not a valid Host name (8 characters or less). This file will not be stored in SCLM.

User Response: If the file needs to be stored as a long language type i.e. converted to a short name then ensure the language type matches the LONGLANG keywords in TRANSLATE.conf (see the administrator).

BWB00126 **Invalid shortname conversion for :** *long name*

Explanation: An error occurred within the translate file in creating a shortname.

User Response: Review the operations log for further errors. Report this problem to the systems programmer or SCLM administrator.

BWB00127 **Check access permissions on translate table :** *file name*

Explanation: Redundant message.

User Response: Ignore

BWB00128 **Error threshold reached in shortname**

Explanation: The error threshold of 5 shortname error conversions has been reached, the migrate process terminates.

User Response: Review the operations log for further errors. Report this problem to the systems programmer or SCLM administrator.

BWB00129 **Member being migrated already exists - not migrated. Member :** *member*

Explanation: An attempt to migrate in a member was made where the member already exists in the group hierarchy and the force option was not used.

User Response: If needed members can be force migrated setting the 'force' flag on from the migrate panel. Warning: this is not advised as normal checkout/checkin processing should be used as existing members will be overlayed.

BWB00130 **Copy failed :** *data set name 1* **to** *data set name 2*

Explanation: A copy of an SCLM member from the HFS to an SCLM dataset has failed.

User Response: The Systems Programmer should check the output dataset is valid and there are no space problems or security violations.

BWB00131 **Check MVS SYSLOG for possible security violation**

Explanation: Informational: extends BWB00130.

User Response: Inform Systems Programmer of this message.

BWB00132 **Error threshold of 20 errors reached in copy**

Explanation: For a migrate or import the error threshold of 20 has been reached for copy failures from HFS to/from SCLM. Processing terminates.

User Response: Review the operations log for further errors. Report this problem to the systems programmer or SCLM administrator.

BWB00133 **Error reading migrate Messages**

Explanation: In Migrate processing an error occurred when reading the Messages dataset. Processing continues but Migrate messages will not be included in the display output.

User Response: Review the operations log for additional error messages.

BWB00134 **SCLM migration successful for the following number of files :** *number of files*

Explanation: Informational

User Response: None

BWB00135 **Migration message data set :** *data set name*

Explanation: Informational

User Response: None

BWB00136 **Migration report data set :** *data set name*

Explanation: Informational

User Response: None

BWB00137 SCLM migration unsuccessful for the following number of files : *number of files*

Explanation: Errors occurred with a number of files during migration processing and are listed here. Migration may have only partially succeeded for some files/members.

User Response: Review the operations log for additional error messages.

BWB00138 Refer to migration message data set : *data set name*

Explanation: Migration messages have been written to this dataset on z/OS for reference.

User Response: Review the migration dataset if required.

BWB00139 SCLM Update error for project archdef : *archdef name*

Explanation: An archdef was selected to add migrating members to but the SCLM update to the archdef has failed.

User Response: Check the archdef selected within SCLM for errors and review the operations log for further errors. Archdef may have to be manually updated with members if required.

BWB00140 Following project ARCHDEF updated : *archdef name*

Explanation: Informational:

Indicated archdef has been updated with migration members.

User Response: None

BWB00141 Error occurred reading ARCHDEF : *archdef name*

Explanation: The archdef member to be updated with migrating members already exists but the read has failed. Migration continues without archdef update.

User Response: Check the validity of the selected archdef. Archdef may have to be manually updated with members if required.

BWB00142 Error occurred updating ARCHDEF : *archdef name*

Explanation: An archdef was selected to add migrating members to but the SCLM update to the archdef has failed.

Bad allocation of archdef member at the development level.

User Response: Check the archdef selected within SCLM for errors and review the operations log for further errors. Archdef may have to be manually updated with members if required.

BWB00143 UNJAR failed on following file : *file name*

Explanation: An error has occurred in unzipping a file using the java JAR command. This function uses java to UNJAR a zip file for transferring files across the network.

User Response: Review the operations log for additional error messages.

The Systems Programmer should ensure a valid java path exists in httpd.env and check for space problems within the HFS directory WORKAREA.

BWB00144 Check CGI-VCMPATH variable in httpd.env with your administrator

Explanation: The VCMPATH specified in the httpd.env file may not be correct.

The Environment variable CGI_VCMPATH determines the HOME directory path where the configuration files reside for SCLM Developer Toolkit.

This was determined by the install directory specified in install job BWBINST1. By default /var/SCLMDT.

User Response: Systems programmer to investigate customization.

BWB00145 J2EEPUT in httpd.conf must be CGI-VCMPATH/WORKAREA/*

Explanation: The httpd.conf directive must be correct for J2EEPUT.

User Response: Systems programmer to investigate customization

BWB00146 Error reading listing file : *file name*

Explanation: Files to be migrated are contained in a HFS listing file. An error occurred while reading this file and migration processing terminates.

User Response: Systems programmer should check HFS file permissions for specified user and available space.

BWB00147 Error in ASCII/EBCDIC conversion : *file name*

Explanation: An error has occurred during an ASCII to EBCDIC file conversion. DT uses the iconv routine within HFS to codepage convert.

User Response: Systems programmer should check ASCII/EBCDIC customization parameters in

TRANSLATE.conf. Check file being converted for special characters or empty file.

BWB00148 Invalid hostname structure (directory/filename)

Explanation: Host file structure does not contain a '/' as first character.

User Response: Review operations log to view Host File name passed to transfer routine.

BWB00149 Invalid direction : *direction*

Explanation: The direction keyword is not valid for file transfer. Valid direction keywords are TOHOST or FROMHOST.

User Response: Review operations log for direction keyword passed and retry.

BWB00150 Invalid filetype : *file type*

Explanation: The filetype keyword for transfer is not valid. Valid filetype is BINARY or TEXT.

User Response: Review operations log for filetype keyword passed and retry.

BWB00151 ZIP option only valid for TOHOST transfer

Explanation: ZIP option (ZIP = yes) is only valid for transfer direction TOHOST.

User Response: Retry operation without ZIP flag set if transferring from host.

BWB00152 ZIP option only valid for BINARY transfer

Explanation: ZIP = YES has been set for transferring a file other than of filetype BINARY. This is not valid.

User Response: Retry operation without ZIP flag set if transferring text file.

BWB00153 Check valid host name or file permissions

Explanation: Accompanies MSG BWB00130.

Copy has failed.

User Response: Review that the file permissions on the HFS directories or Host file name are valid.

BWB00154 Unable to locate host directory : *host directory*

Explanation: The host directory is part of the HOSTFILE name specified. The HFS directory does not exist.

User Response: Systems programmer should check the HFS directory name specified.

BWB00155 UNZIP Failed on following file : *host file*

Explanation: An unzip has failed on the transferred file. SCLM DT uses the java JAR command to unzip (unjar) this file.

User Response: The Systems Programmer should ensure a valid java path exists in httpd.env and check for space problems within the HFS directory WORKAREA.

BWB00156 ZIP File copied & unzipped into directory : *directory*

Explanation: Informational

User Response: None

BWB00157 Unable to delete HFS file : *file name*

Explanation: The cleanup routine to delete HFS workarea files after operation has completed has failed.

User Response: Systems programmer should check listed file and delete/remove manually if desired.

BWB00158 Select DETAILS>> button for build messages, reports, listings

Explanation: Informational

User Response: Select DETAILS to view build messages, reports and listings.

BWB00159 Select DETAILS>> button for promote messages and reports

Explanation: Informational

User Response: Select DETAILS to view build messages, reports and listings.

BWB00160 Failed read of migrate job dataset

Explanation: The temporary skeleton file that is used for creating the batch job could not be read: USERID.TEMP.DWSKEL.

Migration processing terminates.

User Response: Review the operations log for additional error messages.

BWB00161 * Temporary recover data set allocation failed *****

Explanation: In version processing a temporary recovery dataset: USERID.VCM.TEMP.RECOVER could not be allocated.

User Response: Systems programmer should review

the MVS log for any dataset allocation failures.

BWB00162 ANT build failure

Explanation: In a Java or J2EE build an error has occurred running the ANT script in HFS. This may just be Java compile errors.

User Response: Review the ANT build listing returned in the operations log to resolve any build errors.

BWB00163 Review ANT Listing in Log

Explanation: Informational

User Response: Review listing.

BWB00164 Review failure - Build continues

Explanation: A copy of a member from an MVS dataset to a HFS file has failed. Build processing continues.

User Response: Review the operations log for further error messages.

BWB00165 Member account record deleted from development group

Explanation: During a checkout (lock) an SCLM member account record was created in the development group. The unlock or lock failure results in the account record being deleted.

User Response: None

BWB00166 ADD new member cancelled as member exists at group : *group*

Explanation: An add new member was attempted but the member already resides up the group hierarchy chain. Processing terminates.

User Response: None

BWB00167 Error in shortname translation

Explanation: An error has occurred in translating the longname file to an SCLM host shortname.

User Response: Review the operations log for further error messages.

BWB00168 Ensure check-in member was not an empty file

Explanation: A codepage translation error occurred when saving a file/member. Ensure the file saved was not an empty file.

User Response: Check file not empty otherwise review operations log for further error messages.

BWB00169 Member being migrated already exists - not migrated. Member : *member*

Explanation: A member was selected to be migrated but already exists in the group hierarchy chain.

User Response: Use normal EDIT/checkout processing, otherwise use the migrate command with the force flag set. Warning this is not advised as previous versions of the member may be overwritten.

BWB00170 Error threshold of 5 errors reached in ASCII/EBCDIC conversion

Explanation: An error threshold of 5 members being migrated has resulted in codepage conversion errors. Migrate processing terminates.

User Response: Review the operations log to determine errors with migrated members.

BWB00171 No members selected to copy and migrate

Explanation: There were no valid members selected to be migrated into SCLM. Processing terminates.

User Response: If this is an error then review the operations log for further error messages.

BWB00172 Promote has Failed - Check Promote messages

Explanation: The promote has failed. Promote messages will be returned in the log or reside in the promote messages dataset if specified in the request.

User Response: Review the failing messages.

BWB00173 ERROR - Unable to allocate temp translate data set

Explanation: For migrate and import the files requiring long/short name conversion are written to a temporary sequential dataset DD LSTRNPRC of attributes SPACE(1,1) RECFM=VB LRECL=1036 UNIT=VIO. This dataset could not be allocated. Processing terminates.

User Response: Systems Programmer should check the MVS SYSLOG for any dataset allocation errors or violations.

BWB00174 ERROR - In translate table routine : FLMLSTRAN

Explanation: A shortname longname translation has occurred calling the translate module FLMLSTRN. Additional error messages will be displayed.

User Response: Review the operations log for additional error messages and reasons provided.

BWB00175 No associated longnames will be displayed

Explanation: A previous error in retrieving longnames has occurred and any longnames associated with an SCLM host shortname will not be displayed in the project view.

User Response: Review the operations log for other error messages pertaining to translation services.

BWB00176 WARNING in EBCDIC/ASCII conversion. Unprintable characters or empty file

Explanation: An EBCDIC to ASCII codepage translation error has occurred. Likely causes are unprintable special characters or an empty file.

User Response: View the member being translated. Special characters or null characters (X'00') may cause this error when browsing/editing a file.

Bibliography

Publications referred to in this document:

HTTP Server Planning, Installing and Using, SC31-8690

P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook, SG24-4757

z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide, SC34-4817

z/OS ISPF Software Configuration and Library Manager Reference, SC34-4818

z/OS Information Roadmap, SA22-7500

z/OS UNIX System Services Command Reference, GA22-7802

z/OS UNIX System Services Messages and Codes, GA22-7807

z/OS UNIX System Services Planning, GA22-7800

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.

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
Mail Station P300
2455 South Road
Poughkeepsie New York 12601-5400
U.S.A.

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR

PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both.

AIX
Domino
IBM
Library Reader
Lotus
MVS
OS/390
RACF
Redbooks
S/390
WebSphere
z/OS

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

Linux is a 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.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

Index

Special characters

A

B

C

E

F

G

H

I

J

L

M

O

P

R

S

T

U

W

Z

Program Number: 5655M9900

Printed in USA

SC31-6899-01