IBM Cúram Social Program Management
Version 7.0.2

*Cúram Third-Party Tools Installation
Guide for Windows*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 21

Contents

List of Figures

Chapter 1. Installing Third-Party Tools for Windows

You must install and configure software from third parties to use IBM Cúram Social Program Management. Installation and postinstallation instructions are provided for the various supported third-party products for a Windows installation.

Introduction

Installation Overview

To be able to use the IBM Cúram Social Program Management it is necessary it is necessary to install and configure software from third parties. This section describes how to install the third-party products and carry out the post-installation setup tasks.

Third-Party Products Overview

The following document outlines the products that are needed and, for each product, whether post-installation configuration is required.

If you are absolutely happy that you have installed all of the third-party products correctly and performed the necessary post-installation tasks, perhaps because you are working in an environment where your computer has been ghosted, you can skip this document.

Installing Third-Party Software Products

Overview

This chapter describes the software products you need to install before installing IBM Cúram Social Program Management.

Important: IBM Cúram Social Program Management will definitely not run unless you follow the instructions; also, in some cases there is information in this document about product configuration that you will not find anywhere else (including the product documentation).

For each product it tells you:

- Briefly what the product is and why it is used;
- Installation instructions (in most cases, this involves simply following the defaults provided by the product installer); and
- Any post-installation tasks required.

This document does not provide specific instructions for installing fix packs unless the fix pack installation has some product or application-specific steps involved.

DBMS Installation

Both DB2 and Oracle are supported as database servers. DB2 is the name used on Microsoft Windows and UNIX platforms; on IBM z/OS® the name is IBM DB2® for z/OS . Both DB2 and Oracle may be installed on a given machine but typically it is necessary to install only one of the databases.

The H2 database is supported as a development database.

Note: No particular character set is recommended for installation and setup of the DBMS. The character set configured should be appropriate for the character range that will be used in the application. Please refer to the product sections below for further details.

Installing IBM DB2 for Linux, UNIX, and Windows
Ensure that your account has administrative privileges and then follow the DB2 installer instructions to complete a default installation. You do not need to manually create a DB2 database. The platform software provides Ant scripts that you can run as a postinstallation step to create a basic test database.

Note the following options that are presented during a default installation:

- The *Name* and *Password* of the administrator account. Use an account and password as per the standards and requirements of your site and DB2. If it is an existing user, that user must be a member of the Administrator group. The informational message about `OLE DB support component` can be safely ignored.
- Certain editions of the DB2 installer support federated databases. If the installer presents an option that is defaulted to **This machine will be the instance-owning database partition server**, then change this option to **This machine will be a single-partition database server**.
- You must choose MBCS or SBCS, depending on your requirements. If you are unsure of what database encoding option to select, see the related information about data encoding.

DB2 Database Encoding

This section provides important information about issues with DB2 and DB2 for z/OS database encoding, related sizing information, and action you need to consider taking.

What is the Issue?

When using a multi-byte character set (MBCS) and/or encoding DB2 processes columns with respect to their byte size, not their character length. This means that a CHAR, VARCHAR, or CLOB column, when using multi-byte characters, may store fewer characters, depending on the actual character length(s), than the column length specification indicates.

Consider the following illustration:

- A CHAR or VARCHAR column modeled with a length of 16.
- The sixteen-character string, "Marge says hello", (without accented characters) requires 16 bytes for storage in a single-byte character set (SBCS).
- A similar sixteen-character string, but with accented characters, "Márge says héllo", requires 18 bytes for storage in UTF-8, a multi-byte character set (MBCS).

In the case of the single-byte data the string will fit and processing will be successful; but, in the case of the multi-byte data the string will not fit, resulting in overflow errors at run time. That is, normally an IBM Cúram Social Program Management web client will capture and report field size errors in a user-friendly manner. But, in a case as above, because it checks the number of characters and not the byte length, the client will not identify this size mismatch, causing the user to receive an "un-handled server exception" error, which is an underlying SQL Code -302 error.

How Cúram Addresses the Issue

Cúram provides modeling and build-time capabilities to resize its database columns to address the issue above. These capabilities are described further in the *Cúram Modeling Reference Guide* and *Cúram Server Developer's Guide*.

Because Cúram provides support for multiple languages out-of-the-box its support for MBCS data is enabled by default with the maximum expansion set. These expansion settings are appropriate to ensure that new users, testing environments, etc. do not encounter any errors due to their language, encoding, and database sizing. Also, users may find they require MBCS data when they import or copy/paste data from other applications into their Cúram system. However, these defaults may not be appropriate for all environments. The following section describes some considerations for changing these expansion settings.

What You Need to Consider

It is very important to carefully consider your data encoding requirements with respect to DB2 and Cúram in order to avoid unexpected behavior with how the database stores characters.

The preceding illustration represents a boundary case in that the data length matches the maximum column width. In many cases it's unlikely that even with MBCS characters that an overflow situation will occur since most data doesn't reach the maximum defined size; however, you do need to be prepared for the possibility of these error situations.

You should use the database character set encoding appropriate to your application and environment. If possible, you should consider using an SBCS and encoding that supports your requirements. For example, CP1252 supports most Western European characters. However, CP1252 (or other SBCS encodings) may not support characters coming from different or "broader" character sets/encodings (e.g. UTF-8) that users may be used to copying and pasting into their browser for Cúram.

At the point of installing your DB2 (or DB2 for z/OS) database you only need to identify your requirement for SBCS or MBCS data and be prepared to take appropriate action before building your Cúram database:

- If you require characters that use multiple bytes then you need to consider whether the default Cúram settings are appropriate. The necessary database space is dependent on various factors including:
  - The specific character sizes - in DB2 (and DB2 for z/OS) MBCS data can range from 1 to 4 bytes.
  - The frequency of MBCS characters, which can depend on the application, language, locale, column usage within the application, etc.
  - The information density of the language and locale. For instance, while some languages may require more bytes per character, each character may represent more information than, for instance, an alphabetic character and may fit into a field without any size adjustment.

  See the *Cúram Server Developer's Guide* section, *Planning for DB2 MBCS Data*, for more information on MBCS data sizing considerations.
- If an SBCS is adequate then you should plan to disable database expansion as described in the "Planning for MBCS Data" topic in the *Cúram Server Developer's Guide*.

Post-installation

Later versions of DB2 do not include tooling such as Control Center. If your DB2 version includes Control Center you can follow the steps in the Creating and configuring a database using Control Center section below. Otherwise, convenience scripts are provided for creating a basic test database and you can follow the steps in the Creating and configuring a database using scripts section, also below.

Creating and configuring a database using Control Center

1. Go into the General Administration Tools/Control Center.
2. Select theDatabases folder (All Systems... <Hostname>... Instances... DB2... Databases).
3. Right click the mouse.
4. SelectCreate Database... Standard.
5. There are six pages and a summary to the wizard:
6. On page one, fill in the database name (alias should be the same as the name);

After creating the database additional parameters need to be configured. There is no command line to perform this so the Configuration Assistant and the Control Center must be used:

1. Go into the General Administration Tools/Control Center.
2. Select theDatabases folder (All Systems... <Hostname>... Instances... DB2... Databases).
3. Select the database that was created in the previous step.
4. Right click the mouse.
5. SelectConfigure Parameters.

6. Update theLOCKTIMEOUT - Set Lock timeout (in theApplications section) to at least1 second, or any desired higher value.
7. Reboot the PC or cycle all DB2 services. To cycle all DB2 services, please do the following from within a DB2 command prompt:
   - connect reset
   - force application all
   - db2stop
   - db2start

After configuring the database you must create the tablespaces required for the storage and running of the application.

This can be done using the provided Apache Ant scripts by invoking:

```
ant -f %CURAMSDEJ%\util\db2_postconfig.xml -Ddb2.dir=<DB2_directory>
```

where <DB2_directory> is the DB2 installation path (e.g. <drive>:\IBM\SQLLIB). Otherwise, complete the following steps:

1. Start a db2cmd window by running the db2cmd command at the command prompt.
2. Enter the following at the command prompt:

   ```
   db2 connect to <database_name> user <user_name> using <password>
   db2 CREATE BUFFERPOOL highmem SIZE 50 PAGESIZE 32K
   db2 connect reset
   ```
3. Reboot the PC or cycle all DB2 services. Please see above for details on how to cycle all DB2 services.
4. Start a db2cmd window.
5. Enter the following at the command prompt:

   ```
   db2 connect to <database_name> user <user_name> using <password>
   db2 create tablespace Curam_L PAGESIZE 32K MANAGED BY SYSTEM using (<dir_name_A>) BUFFERPOOL highmem
   db2 create temporary tablespace Curam_T PAGESIZE 32K MANAGED BY SYSTEM using (<dir_name_B>) BUFFERPOOL highmem
   db2 connect reset
   ```
6. Reboot the PC or cycle all DB2 services. Please see above for details on how to cycle all DB2 services.

Note: <dir_name_A> and <dir_name_B> should be non-existent directories, i.e. directories that don't exist before the command is run - on a drive with a significant amount of space.

All DB2 commands above are one-line commands, i.e. no line breaks.

If the username used to connect to the database when executing the commands is not the same as the username which will be used to access the database, the following command should also be executed, where <user_name> is the name of the user that will be used to access the database:

```
db2 grant use of tablespace Curam_L to user <user_name> with grant option
```

Creating and configuring a database using scripts

Ant scripts are provided for creating and configuring a basic test database. To create a database, invoke the following commands, which use the database properties from your Bootstrap.properties file:

- `ant -f %CURAMSDEJ%\util\db2_createdb.xml`
- `ant -f %CURAMSDEJ%\util\db2_postconfig.xml -Ddb2.dir=<DB2_directory>`

  where <DB2_directory> is the DB2 installation path (e.g. <drive>:\IBM\SQLLIB).
- `ant -f %CURAMSDEJ%\util\db2_createdb.xml restart.db2`

Note: this script restarts your DB2 system.

- `ant -f %CURAMSDEJ%\util\db2_optimizedbrecreation.xml`

The following script invocation can be used to drop the database to rerun the process above:

`ant -f %CURAMSDEJ%\util\db2_createdb.xml dropdb`

Providing a DB2 License File

This post-installation step is required for all users of IBM DB2 for Linux, Unix and Windows for the IBM Cúram Social Program Management Platform development or runtime environments.

An empty `db2jcc_license_cu.jar` file exists in the `%CURAMSDEJ%\drivers` directory to allow for Eclipse classpath dependencies in the CuramSDEJ project. This empty jar should be overwritten with a real license for accessing IBM DB2 for Linux, Unix and Windows.

The IBM DB2 `db2jcc_license_cu.jar` file should be copied from `<DB2_directory>\java\db2jcc_license_cu.jar` (where `<DB2_directory>` is the DB2 installation path; e.g. `C:\IBM\SQLLIB`) to `%CURAMSDEJ%\drivers` (where `%CURAMSDEJ%` points to the root CuramSDEJ location).

Replacing the Packaged DB2 Drivers

Usually the latest JDBC drivers available at the time of release are packaged with IBM Cúram Social Program Management. However, if you wish to replace the drivers shipped in `%CURAMSDEJ%\drivers`, copy the following files from `<DB2_directory>\java`.

(where `<DB2_directory>` is the DB2 installation path e.g. `<drive>:\IBM\SQLLIB`)

- db2jcc.jar
- db2jcc_license_cu.jar
- sqlj.zip

Replace the drivers `db2jcc.jar`, `db2jcc_license_cu.jar`, and `sqlj4.zip` in `%CURAMSDEJ%\drivers` directory.

Using DB2 pureScale

When using DB2 pureScale® with Cúram you will need to perform the following steps to set the necessary data source property or properties when using DB2 from the command line (e.g. with Cúram batch processing). See the relevant Information Centers for DB2 and WebSphere for their specific pureScale settings.

You must generate a `.bindings` file based on your `Bootstrap.properties` file database settings, which specify the DB2 pureScale connect member. To do this:

1. In your `Bootstrap.properties` file set property `curam.db.enable.bindings.generation=true` and specify a valid location value for property `curam.environment.bindings.location`; e.g., `curam.environment.bindings.location=C:/Temp;`
2. Run the Ant `configtest` target, which will cause the `.bindings` file to be generated in the specified location;
3. In your `Bootstrap.properties` file remove `curam.db.enable.bindings.generation=true` or set it to `false` and set `curam.db.disable.bindings.generation=true`;
4. Set the Content value for the relevant pureScale data source properties in the `.bindings` file (much easier if you sort it first); for example, set `enableSysplexWLB` to 'true'. Save the changes.

From this point forward the Cúram DB2 data source, when used from the command line, will use these properties and changes to the database properties in `Bootstrap.properties` would need to be reflected in `.bindings` or by rerunning the procedure above. However, Ant scripts using the `<sql>` task will not use these pureScale settings. These Ant scripts (e.g. `database` target) are typically not run

frequently nor have a processing profile that would require pureScale settings; but, you can modify scripts as needed to specify these properties using the Ant `<connectionProperty>` nested element.

Configuring for circular transaction logging
When you use a database with circular transaction logging enabled, certain transactions can exceed the available log file space and fail. To avoid this issue, either use archive logging or set the available log size and quantity appropriately until it meets the needs of the transaction.

About this task

A common point for this failure is when the `prepare.application.data` Ant target is running, as this target publishes all the CER rule sets on the system. This Ant target is typically run after a clean database build. If the log is too small, it can result in an `SQLCODE -964` error.

You can use the following example to help you to increase the DB2 log file size and quantity. The exact amount of log file storage that is required varies from system to system. For more information about increasing the number and size of the log files available, see the specific documentation for your database.

Procedure

1. Open a command prompt and enter db2cmd.
2. Enter the following command:

   ```
   db2 connect to db_name user db_user_name using db_password
   ```

   Where *db_name*, *db_user_name*, *db_password* are the credentials of the database.
3. Enter the following commands:

   ```
   db2 update db cfg for db_name using logfilsiz log_file_size
   ```

   ```
   db2 update db cfg for db_name using logprimary primary_log_files
   ```

   ```
   db2 update db cfg for db_name using logsecond secondary_log_files
   ```

   Where the temporary values are as follows:

   - The log file size. Set *log_file_size* to 1024.
   - The number of primary log files. Set *primary_log_files* to 50.
   - The number of secondary log files. Set *secondary_log_files* to 100.
4. Restart the database by entering the following commands:

   ```
   db2stop
   db2start
   ```

Remote DB2 for z/OS Connectivity

About this task

Before the connection can be established to the remote database it must be configured. The full details of installation of DB2 for z/OS are beyond the scope of this document set. However the following are the main post-installation steps to note:

Procedure

1. A database can be configured for EBCDIC, ASCII, or UNICODE mode for use by the application. This can be done when creating the database using theCCSID keyword. For ASCII or UNICODE databases

see the *Cúram Server Developer's Guide* for information about required property `curam.db.zos.encoding`.

```
CREATE DATABASE <database_name> BUFFERPOOL BP0 INDEXBP BP0 STOGROUP
<storage_group> CCSID <EBCDIC, ASCII or UNICODE>;
```

2. An environment variable called DB2JCC_LICENSE_CISUZ_JAR must be created that points to the installed DB2 for z/OS license jar file used for connectivity to the remote database server. This is normally named `db2jcc_license_cisuz.jar` and is provided with DB2 for z/OS or DB2 Connect.

Oracle database

Oracle database is supported as a database server.

Note: It is possible to use a remote database by using the Oracle Type 4 Driver that is supplied with the SDEJ.

Installing the Oracle database

Assuming that no previous versions of Oracle are installed, you can complete a typical Oracle database server installation.

Oracle Database Encoding

It is important to consider the character set for data that will be stored on your database when configuring your database for use with IBM Cúram Social Program Management. For Oracle, there are 2 parameters to consider: NLS_CHARACTERSET and NLS_LENGTH_SEMANTICS.

- The NLS_CHARACTERSET parameter details the allowable character set of any data loaded to the database, generally AL32UTF8 is recommended by Oracle here.
- The NLS_LENGTH_SEMANTICS determines how Oracle interprets length specifiers on CHAR and VARCHAR columns. To handle supplementary characters, e.g. ß in German, where the storage of the character would be two bytes and could overrun the length of a defined column; the NLS_LENGTH_SEMANTICS parameter should be set to CHAR as this directs the database to size columns with a character length rather than byte length.

Replacing the Packaged Oracle JDBC Drivers

Usually the latest JDBC drivers available at the time of release are packaged with IBM Cúram Social Program Management. However, if you wish to replace the drivers shipped in `%CURAMSDEJ%\drivers`, please follow the sample steps below:

- Copy `<ORA_directory>\product\12.1.x\dbhome_1\jdbc\lib\ojdbc6.jar` to a `<temp>` location.

  where `<ORA_directory>` is the Oracle installation path (e.g. `<drive>:\oracle121`)
- Rename `ojdbc6.jar` to `ojdbc.jar` (to match the name of the driver shipped).
- Copy `<ORA_directory>\product\12.1.x\dbhome_1\oc4j\sqlj\lib\translator.jar` to a `<temp>` location.
- Copy `<ORA_directory>\product\12.1.x\dbhome_1\oc4j\sqlj\lib\runtime12.jar` to a `<temp>` location.
- Replace the drivers `ojdbc.jar`, `runtime12.jar`, and `translator.jar` in `%CURAMSDEJ%\drivers` with the drivers in your `temp` location.

Please note, the renaming of the ojdbc6.jar is necessary due to the fact that the SDEJ expects the driver file to be named `ojdbc.jar`.

Post-installation

There are a number of post-installation tasks which must be performed:

- Create an Oracle role for the application server(s).

Note: The post-installation tasks require connecting to Oracle as the privileged 'sys' user. Immediately after installing Oracle, the password for this user is *'change_on_install'*. Oracle requires that this be changed.

Create an Oracle role for Application Servers

The application needs certain privileges in order to use the Oracle XA interface. Later, when configuring the application, the username under which the server connects to Oracle is specified. The appropriate privileges must be assigned to this username for the server to work successfully.

An easy way to bundle together the various privileges required is to create an Oracle *Role*. Privileges can be granted to this role. Later this role can be granted to your users, thereby granting all the privileges associated with that role.

The following commands create a role called CURAM_SERVER and give it the necessary privileges. A user named CURAM_USER is then assigned that role and given the password PASSWORD. The commands should be run inside an Oracle SQLPlus window.

Note: To run the commands from SQLPlus type the following at a command prompt `sqlplus ?/? as SYSDBA`.

```
CREATE ROLE "CURAM_SERVER";

GRANT RESOURCE TO "CURAM_SERVER";

@%ORACLE_HOME%\RDBMS\ADMIN\xaview.sql

GRANT SELECT ON V$XATRANS$ TO PUBLIC;

GRANT SELECT ON PENDING_TRANS$ TO PUBLIC;

GRANT SELECT ON DBA_2PC_PENDING TO PUBLIC;

GRANT SELECT ON DBA_PENDING_TRANSACTIONS TO PUBLIC;

GRANT EXECUTE ON DBMS_SYSTEM TO CURAM_SERVER;

CREATE USER <CURAM_USER> IDENTIFIED BY <PASSWORD> DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP";

GRANT "CONNECT", "CURAM_SERVER", UNLIMITED TABLESPACE TO <CURAM_USER>;

ALTER SYSTEM SET "_optimizer_ansi_join_lateral_enhance"=false;
```

Figure 1: Oracle Configuration

Note: Replace <CURAM_USER> and <PASSWORD> in the CREATE USER command with the username and password you wish to use for the database user.

Redo Log

Certain IBM Cúram Social Program Management transactions perform significant insert activity and could be impacted by available redo log space. The most likely situation for this would be during invocation of the Ant `prepare.application.data` target, generally run after a clean database build, as this is publishing all the CER Rule Sets for Cúram. Refer to the Oracle documentation on allocating the appropriate size for the redo logs. The exact amount of activity and required redo log space will vary from system to system.

Configuring for circular transaction logging
When you use a database with circular transaction logging enabled, certain transactions can exceed the available log file space and fail. To avoid this issue, either use archive logging or set the available log size and quantity appropriately until it meets the needs of the transaction.

A common point for this failure is when the `prepare.application.data` Ant target is running, as this target publishes all the CER rule sets on the system. This Ant target is typically run after a clean database build.

For information about increasing the number and size of the log files available, see the specific documentation for your database. The exact amount of log file storage that is required varies from system to system.

H2

Overview

H2 is an SQL database engine written in Java™ that implements the JDBC API. A browser-based console application is included.

Limitations in the support of H2:

- Usage of this database is only for development.
- It is not supported at runtime.
- EAR files cannot be built for this database.
- The `configure` target which automatically configures the application server cannot be run while this database is in use.

Note: For more information on build files and their targets consult the *Cúram Server Developers Guide*.

Installation

H2 does not require any specific installation steps. The database is self contained within the delivered file: `%CURAMSDEJ%\drivers\h2.jar`.

Configuration

To use an H2 database you must set the `Bootstrap.properties` file to specify the set of properties relevant to H2 for obtaining a connection to the database, e.g.,

```
curam.db.type=h2
curam.db.name=curamdb
curam.db.username=curam
curam.db.password=qqnscP4c4+s=
# H2 directory.
# Default is home directory
# (i.e. C:/Documents and Settings/<username>). (Optional)
curam.db.h2.directory=C:/H2
# Mode remote|embedded
curam.db.h2.mode=embedded
# For remote mode also specify:
curam.db.serverport=9092
curam.db.servername=localhost
# Lock Time Out in ms. Default is 1000, i.e. 1 second. (Optional)
curam.db.h2.locktimeout=20000
# Property to disable MVCC. Default: true. (Optional)
curam.db.h2.mvcc=true
```

Figure 2: Bootstrap.properties

Once you have updated `Bootstrap.properties` and rebuilt the server and database you can develop in exactly the same way as you would with Oracle or DB2.

For more information on `Bootstrap.properties` consult the *Cúram Server Developers Guide*.

H2 Modes

The following H2 modes are supported for application development:

- Embedded - in embedded mode, an application opens a database from within the same JVM using JDBC. This is the fastest and easiest connection mode. The disadvantage is that a database may only be open in one virtual machine (and class loader) at any time.

- Remote - in remote mode (sometimes called client/server mode), an application opens a database remotely using the JDBC or ODBC API. Many applications can connect to the same database at the same time. The remote mode is slower than the embedded mode, because all data is transferred over TCP/IP.

You can specify the required mode by using the following property in `Bootstrap.properties` file:

```
# Mode remote|embedded
curam.db.h2.mode=embedded
```

Multi-Version Concurrency Control (MVCC)

The MVCC feature allows higher concurrency than using exclusive table level or row level locks. When using MVCC in this database, delete, insert and update operations will only issue a shared lock on the table. An exclusive lock is still used when adding or removing columns, when dropping the table, and when using `SELECT... FOR UPDATE`. Connections only 'see' committed data, and their own changes. That means, if connection A updates a row but hasn't committed the change, connection B will see the old value. Only when the change from connection A is committed, the new value is visible to other connections (read committed). If multiple connections concurrently try to update the same row, the database waits until it can apply the change, but at most until the lock timeout expires.

By default, MVCC is set to `true`.

You can switch off MVCC by using the following property in `Bootstrap.properties` file:

```
# Property to disable MVCC. Default: true. (Optional)
curam.db.h2.mvcc=false
```

H2 Web Console

When you start H2 by running the `org.h2.tools.Server` class in `h2.jar` its web console is started and can be accessed at the URL:

```
http://localhost:8082/
```

The JDBC connection URL you specify in the login screen is based on the curam.db.name, curam.db.username, and curam.db.h2.directory values in `Bootstrap.properties`, which define the database name, SCHEMA name, and the database location in the file system. So, if your database name is `curamdb`, your username is `curam` and curam.db.h2.directory is defaulting to your home directory, your JDBC string would look like this:

```
jdbc:h2:tcp://localhost/~/curamdb;schema=curam;FILE_LOCK=SOCKET
```

Note, if, for example, curam.db.h2.directory is `C:/H2`, then your JDBC string would look like this:

```
jdbc:h2:tcp://localhost/file:C:/H2/curamdb;schema=curam;FILE_LOCK=SOCKET
```

Specify the values for **User Name** and **Password** as in your Bootstrap.properties and then click the **Connect** (or **Test Connect**) button. Once connected you'll have an SQL text control, etc. available.

Further Reading

For more information about H2, see the H2 manual.

Related information
H2 Database Engine manual

Apache Ant

Installing Apache Ant
Apache Ant from the Apache Jakarta project is a build tool that is based on Java.

Before you begin
You do not need to extract the Apache Ant compressed file into a directory called `ant`. The file extracts to the `apache-ant-version` directory.

Procedure

1. Download the Ant compressed file from the Apache website.
2. Extract the file to a directory of your choice on your computer
   For example, extract `apache-ant-version-bin.zip` to `C:\apache-ant-version`.
   The installation is now complete.

Configuring Apache Ant
You must create Microsoft Windows environment variables and update your Microsoft Windows path for Apache Ant.

Procedure

1. Create an ANT_HOME system environment variable with the value set to the Apache Ant installation directory.
2. Add `%ANT_HOME%\bin` to the PATH environment variable.
3. Create an ANT_OPTS system environment variable with the value -Xmx756m.

Rational Software Architect

Overview

IBM Rational® Software Architect is an Eclipse -based UML modeling tool that is required to do server development. More details on the use of Rational Software Architect can be found in the *Working with the Cúram Model in Rational Software Architect* document and more information on modeling can be found in the *Cúram Modeling Reference Guide*.

Installation

The exact installation steps may vary by the edition and/or version of your software and the relevant IBM documentation should be consulted in addition to the information in this guide.

The IBM Installation Manager must be installed before installing Rational Software Architect and can be installed taking the default options.

Install Rational Software Architect:

1. Start the IBM Installation Manager (**Start > Programs > IBM Installation Manager > IBM Installation Manager**);
2. From the **File** menu select **Preferences** and on the Repositories page add a new entry appropriate to your site that points to your Rational Software Architect installation location and initial installation file and make this the only selected repository;
3. Click **OK** to go back to the main screen;
4. Select **Install** and you should see the appropriate version as an available install package. Select this and click **Next**;
5. Accept licenses, which is dependent on your local license configuration;
6. Choose a location for the Shared Resources;
7. Choose a Package installation location;
8. Choose not to extend an Eclipse version; (If you wish to use Rational Software Architect in an existing Eclipse instance see the product documentation; i.e. https://www.ibm.com/developerworks/

9. On the "Install Packages - select features to install" panel, you need to additionally install: 'Rational Rose Model Import' in order to install the IBM Rational Rose profile required by migrated models in Eclipse;

10. Continue and choose the defaults for the rest of the installation.

Post-installation

You must install the shipped plugins to enable modeling support:

1. Create a `dropins` directory in your Rational Software Architect install location; e.g., `C:\Rational\SDP\dropins`;

2. Create an `rsa_plugin.link` file that has the contents referencing the location where the plugin for Rational Software Architect is delivered to (note: use forward slashes); e.g.:

   `path=C:/Curam/CuramSDEJ/rsa`

3. Edit the shortcut that starts Rational Software Architect to pass the `-clean` option so that the plugins are picked up e.g.

   `...\eclipse.exe -clean -product com.ibm.rational....`

4. Start or restart Rational Software Architect.

The configuration of your Rational Software Architect license is dependent on your local site configuration.

Application Server

The supported application servers on Windows are WebSphere Application Server and WebLogic Server. An application server must be installed on a machine that builds applications for deployment as well as on machines which host the deployed application.

Note: None of the application servers should be installed on a machine which has an underscore in the machine name.

WebSphere Application Server

Installation

Install WebSphere Application Server from the IBM installation media. The default installation is used, without any configuration changes.

If installing WebSphere Application Server as a service, a user account should be created in advance so that it can be used as the credentials for the service. This user account must have administrator privileges.

You should *not* install WebSphere Application Server to a directory that contains spaces in the name, in other words do not use the default `Program Files` directory.

Note: Please avoid installing the WebSphere Application Server sample applications. This is because the sample application Apache Derby DataSource will result in a classpath conflict with the application web client's use of Derby.

Setting the WebSphere Application Server environment variable
Set the required Microsoft Windows environment variable for WebSphere Application Server.

Procedure

Set the WAS_HOME environment variable to the server directory of the WebSphere Application Server installation.
For example, `C:\WebSphere\AppServer`

WebSphere Application Server Network Deployment

Overview

IBM's WebSphere Application Server Network Deployment is an optional component that manages one or many IBM WebSphere Application Server editions and offers advanced deployment services, including clustering, edge services and high availability for distributed configurations.

Installation

Install WebSphere Application Server Network Deployment from the installation media. The default installation is used, without any configuration changes.

If installing the Deployment Manager as a service a user account should be created in advance so that it can be used as the credentials for the service. This user account must have administrator privileges.

You should *not* install WebSphere Application Server Network Deployment to a directory that contains spaces in the name, in other words do not use the default `Program Files` directory.

Note: Please avoid installing the WebSphere Application Server sample applications. This is because the sample application Apache Derby DataSource will result in a classpath conflict with the application web client's use of Derby.

Setting the WebSphere Application Server environment variable
Set the required Microsoft Windows environment variable for WebSphere Application Server.

Procedure

Set the WAS_HOME environment variable to the server directory of the WebSphere Application Server installation.
For example, `C:\WebSphere\AppServer`

WebLogic Server

Installation

When installing WebLogic Server:

1. Click **Next** on the initial welcome screen.

2. Specify the home directory where you wish to install WebLogic Server and click **Next**.

3. Choose a custom installation and click **Next**.

4. De-select all options except the **WebLogic Server** branch and click **Next**;

5. Select the JDK(s) you require and click **Next**;

6. Specify the Product Directory based on the WebLogic Server home directory you specified above and click **Next**;

7. Accept the default to not install as a Windows service and click **Next**;

8. Choose the shortcut location and click **Next**;

9. Review the installation summary and click **Next**;

Once the installation is complete uncheck **Run Quickstart** and click **Done** to exit the installer.

Post-installation

The WLS_HOME environment variable must be setup. It should be set to the `server` directory of the WebLogic Server installation (e.g. `<drive>:\wls121\wlserver\server`).

Java SE/Java EE

Overview

Both a Java SE and a Java EE are necessary.

Installation

Specific installation instructions are not provided for the Java SE or Java EE as each application server ships with a Java SE and Java EE which can be used.

Note: Irrespective of which Java SE and Java EE are used the post-installation steps listed below must be followed.

Post-installation

Procedure

1. An environment variable called JAVA_HOME must be created that points to the installed Java SE. %JAVA_HOME%\bin should be placed at the beginning of the PATH environment variable. On a number of sites, it may be necessary to have multiple Java SE s installed on any given machine. For this reason, the choice of scope for these environment variables (system wide, through a script file or symbolic links) is left up to you.

2. An environment variable called J2EE_JAR must be created that points to the installed Java EE jar file. For WebSphere Application Server this should point at %WAS_HOME%\lib\j2ee.jar. For WebLogic Server this should point at %WLS_HOME%\lib\weblogic.jar; however, note the specific post-install step for using WebLogic Server with Eclipse in "Using a Server Project in Eclipse" on page 16.

   On a number of sites, it may be necessary to have multiple Java EE s installed on any given machine. For this reason, the choice of scope for these environment variables (system wide, through a script file or symbolic links) is left up to you.

Eclipse & Tomcat

Overview

This section describes the steps necessary to setup a machine for development of an application in Eclipse and Apache Tomcat.

Installation

To run in an Eclipse -based development environment the following software is required which can be downloaded from web.

- Eclipse : The IDE used to develop an application.
- Tomcat : The servlet container used to run the client web application. Below are the details of customizations that needs to be done to the Tomcat configuration for the development environment.

  - Modifications to server.xml: It is recommended to change default port of Tomcat due to potential port conflicts with other software. The Connector element in the server.xml file in the <Tomcat_install_dir>/conf directory can be updated, for example to change to 9080: <Connector port="9080" ...>

  - Modifications to context.xml: The <Context> element of context.xml file in the <Tomcat_install_dir>/conf directory should be updated to include a reloadable="true" attribute, i.e. <Context reloadable="true">

  - Modifications to web.xml: The web.xml file in the <Tomcat_install_dir>/conf directory should be updated to include a new keepgenerated init-param. This should be added to the org.apache.jasper.servlet.JspServlet servlet with the value false. See note below. For example:

```
<init-param>
<param-name>keepgenerated</param-name>
<param-value>false</param-value>
</init-param>
```

- Sysdeo Eclipse Tomcat Launcher : An open source Eclipse plug-in which integrates with a Tomcat installation. Install the plug-in by extracting the plug-in zip file to the Eclipse `plugins` directory. The default `eclipse/plugins)` directory is assumed.

- Java SE / Java EE : Oracle's Java SE and Java EE are used when running in Eclipse.

Note:

Tomcat converts JSPs into servlets contained in UTF-8 encoded java sources files by default (for multi-byte character support). These are generated into the `work` folder of the project. The Sysdeo plug-in marks the `work` folder as an Eclipse source folder. If you use the Eclipse build command, the java compiler will expect sytem encoding sources files by default. If any source file in the `work` folder contains non-ascii characters an `Invalid Character` compiler error will be generated and in turn you cannot access the page in a web browser.

The `keepgenerated` prevents Tomcat from saving the source files in the work folder and hence avoid this problem. The Eclipse compiler cannot be changed to compile UTF-8 source files because there is a second source folder called JavaSource which contains file which are not in UTF-8 encoding.

Change this setting does not effect using the application in any way.

The `keepgenerated` flag can be set to true if you wish to view and debug through source files generated by Tomcat, but the problem described above should be considered.

Post-installation

Eclipse - Default Java SE

About this task

Eclipse should be launched with the Java SE installed from the product installation CD. Because many products are Java -based, multiple Java SE installations may already exist on your system. The following steps ensure the correct Java SE is used.

Procedure

1. Eclipse is launched from the `eclipse.exe` executable. Ensure the Java SE installed from the development CD is the first one on the Windows system path when launching Eclipse. An alternative is to use the `-vm` command line argument to the `eclipse.exe` executable. Consult the Eclipse documentation for further details on its command line arguments.

2. Once Eclipse is launched, select **Window** > **Preferences**.

3. On the preferences dialog select **Java**, **Installed JREs** from the tree. The correct Java SE should be selected as the default.

4. Click the **Edit** button.

5. In **Default VM Arguments** enter `-Xmx512M`.

Results

Update Boot Class Path When Using the IBM WebSphere Java SDK: The following XML API jar files (shipped with the SDEJ) should be added to the boot class path if the IBM WebSphere Java SDK is being used.

- `%CURAMSDEJ%\lib\xalan-2.7.1.jar`

- `%CURAMSDEJ%\lib\serializer-2.7.1.jar`

- `%CURAMSDEJ%\lib\xerces-2.9.1.jar`

The boot class path can be set as a JVM parameter. For example:

```
-Xbootclasspath/p: %CURAMSDEJ%\lib\xalan-2.7.1.jar; %CURAMSDEJ%\lib
\serializer-2.7.1.jar;%CURAMSDEJ%\lib\xerces-2.9.1.jar
```

Tomcat

Tomcat with the default configuration incorrectly assumes requests are encoded using ISO-8859-1 instead of UTF-8. This would break string handling in IBM Cúram Social Program Management if request parameters contained UTF-8 extended characters. To avoid this add the `useBodyEncodingForURI="true"` parameter to the `<Connector>` element of the `server.xml` configuration file.

Alter POST limit when using Rules Sets: By default Tomcat limits POST data to 2 MB. This can cause an issue when using Rules Sets, which need to POST data greater than this limit. Although this step is optional, it is recommended for users intending to use Rules Sets functionality. To disable the POST limit in Tomcat add the `maxPostSize="-1"` attribute to the `<Connector>` element of the `server.xml` configuration file. For example:

```
<Connector port="9080" maxThreads="150" minSpareThreads="25"
maxSpareThreads="75" enableLookups="false" redirectPort="8443"
acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
useBodyEncodingForURI="true" maxPostSize="-1"/>
```

Tomcat Plug-in

To use Tomcat from Eclipse the Sysdeo Eclipse Tomcat Launcher requires some configuration:

1. From the menu, select **Window** > **Preferences**.
2. On the preferences dialog select **Tomcat** from the tree.
3. Set the **Tomcat version** to **6.x**.
4. Set **Tomcat home** to the location of the tomcat installation.
5. Under **Tomcat** select **JVM Settings**.
6. In **Append to JVM Parameters** enter `-Xmx512m`.

Using a Server Project in Eclipse

The server project needs to be imported and configured for use in Eclipse.

1. Import the project (`.project`) located in the `CuramSDEJ` folder in the development installation.
2. Import the project (`.project`) located in the `EJBServer` folder in the development installation.
3. Once the projects are imported there are a number of Classpath variables errors. To resolve these navigate to **Windows -> Preferences**, then **Java -> Build Path -> Classpath Variables** and set the following variables.

   - J2EE_JAR - jar file of your Java EE implementation. In the case of WebLogic Server you can generate the `wlfullclient.jar` and set J2EE_JAR to point to it: Run the WebLogic Server JarBuilder tool, by going to the `server\lib` directory (e.g. `cd %WLS_HOME%\lib`) and use the `java` command to create the `wlfullclient.jar` (e.g. `java -jar wljarbuilder.jar`); see the WebLogic Server product documentation for more information on the WebLogic Server JarBuilder tool.
   - JAVAMAIL_HOME - this should point at the folder containing `mail.jar` and `activation.jar` of your Java EE implementation.

The server is launched as a Java process which launches three threads:

- tnameserv: the Transient Name (tname) Server. This facilitates a JNDI look up service for finding resources such as Java classes;
- RMI Server: Server application process;
- JMSLite: JMS Message Engine. See the *Workflow Reference Guide* for more information on JMSLite.

To launch the server, from the `EJBServer/ReferencedLibraries/core.jar` (default package), select the `StartServer` class and select the **Java Application** option from the **Run As** button.

VM Arguments: TheRun Configuration option can be used to manage VM arguments to the process e.g. to pass a higher heap size setting -Xmx756M.

Using a Client Project in Eclipse

The client project needs to be imported and configured to run on Tomcat.

1. Import the project (`.project`) located in the `CuramCDEJ` folder in the development installation.
2. Import the project (`.project`) located in the `webclient` folder in the development installation.
3. Once the projects are imported right-click the project and select **Properties**, then **Tomcat** on the next dialog. Set the following properties:
   - Context Name : Should be set to "/Curam"
   - Can update server.xml file : Selected
   - Mark this context as reloadable: Selected
   - Redirect context logger to Eclipse console : Selected
   - Subdirectory to set as application root : Should be set to "WebContent"
4. Right-click the project again and select **Tomcat Project**, **Update Context Definition**. This updates Tomcat's `server.xml` with an entry for the client application.

The Sysdeo plug-in adds a toolbar to Eclipse for starting and stopping Tomcat. Once it is started, the client application is available at the following URL : http://localhost:9080/Curam/AppController.do

Using the Cúram Business Intelligence and Analytics Viewer project in Eclipse

This section will guide you through the process of configuring the Eclipse environment to view Cúram Business Intelligence and Analytics content. If you have further questions please see the *BIRT Developers Guide* for more details.

1. Import the project into Eclipse by going to **File** > **Import...**. Select **General** > **Existing Projects into Workspace** and click **Next**. Select the root directory to be %CURAM_DIR%\BIApp\CuramBIRTViewer and click **Finish**. %CURAM_DIR% is the Cúram installation directory, which by default is `C:\IBM\Curam\Development`.
2. Select the `CuramBIRTViewer` project in Eclipse, then open the preferences dialog by selecting **Window** > **Preferences** and expand **Java** > **Compiler** and set **Compiler compliance level:** to `1.5`.
3. Select the `CuramBIRTViewer` project in Eclipse, right-click and select **Tomcat Project** > **Update context definition**.
4. Open the preferences dialog by selecting **Window** > **Preferences** and expand **Tomcat** > **JVM Settings** and for the **Classpath** text box select the **Directory** button and add in the full path to the %CURAM_DIR%\EJBServer\project\properties.
5. If you are viewing licensed content via the **Reports** tab (ignore this section if you do not have a **Reports** tab), then Open the preferences dialog by selecting **Window** > **Preferences** and expand **Tomcat** > **JVM Settings** and for the **Classpath** text box select the **Directory** button and add in the full path to the %CURAM_DIR%\Reporting\project\properties. See the *BIRT Developers Guide* for more details.
6. You must now publish all Cúram Business Intelligence and Analytics content to the BIRT Viewer. From the directory %CURAM_DIR%\BIContent execute the command `build client.birt`.
7. Start Tomcat, to verify your viewer is open for business, check `http://localhost:9080/CuramBIRTViewer`.

If you are developing new Cúram Business Intelligence and Analytics content, please see the *BIRT Developers Guide* for more details on how to set up a development environment.

Using Eclipse to Validate the Tabbed Configuration Artifacts

To setup Eclipse to validate the tabbed configuration files with the correct schema open the Eclipse **Preferences** dialog by selecting **Window** > **Preferences** and do the following:

- Select **XML** > **XML Catalog**;
- Click **Add...** to add a new entry;
- For the **Location**, point at the schema file (e.g. `tab.xsd`) in the `%CURAMSDEJ%\lib` directory;
- Leave the rest as defaults and click **OK**;
- Repeat for each of the schema files for the tabbed configuration artifacts. Please refer to the *Cúram Client Developer's Guide* for the full list of schema files. Click **OK** to exit the **XML Catalog** dialog.
- Select **General** > **Editors** > **File Associations**;
- Click **Add...** to add a new entry: `*.tab`;
- Select the new `*.tab` entry and use the **Add...** button to add the XML Editor as the **Associated Editor**;
- Repeat for all the tabbed configuration artifact file extensions. Please refer to the *Cúram Client Developer's Guide* for the full list of extensions.
- Select **General** > **Content Types**;
- Expand **Text** and select **XML**;
- Click **Add...** to enter a file association for XML content and click **OK**. Do this for each of the file extensions.
- Click **OK** to save the preference changes.

Default Text File Encoding

Eclipse allows you to set the default text file encoding at the project level. Changing this from the default is currently *unsupported* for IBM Cúram Social Program Management projects within Eclipse. This does not effect your ability to save files in various encodings on a file by file basis. Consult the *Cúram Server Developer's Guide* and the *Cúram Web Client Reference Manual* for details on file encoding.

Rational Application Developer

Overview

This section describes the steps necessary to setup a machine for development of an application in IBM Rational Application Developer.

Installation

Perform a standard installation of Rational Application Developer.

Post-Installation

Configuration

About this task

The following post installation steps are required to ensure that Rational Application Developer will work with the application.

Set the following Rational Application Developer preferences:

Procedure

1. Switch off build automatically: **Project -> Build Automatically**, and clicking on it, ensuring it is no longer ticked.

2. Switch off validation: **Go to Window -> Preferences -> Validation,** and hit the Disable All button and click Ok.

3. Select **Windows -> Preferences** and then **Java -> Installed JREs** and set the installed JRE to be "WebSphere v8.5 JRE".

4. Set the following file associations via **Windows -> Preferences -> Workbench -> File Associations**:

   • Add *.uim to the file types

   • Add the XML Editor as the associated editor for *.uim.

5. Add the schema to the XML catalog. In **Windows -> Preferences**, select **Web and Xml -> Xml Catalog** and then select "User Specified Entries". Click **Add** and set the following values:

   • Location: <CuramCDEJ>/lib/curam/xml/schema/uim.xsd

   • Key Type: Schema Location

   • Key: file://Curam/UIMSchema.xsd

Results

Update to Boot Class Path: The following XML API jar files (shipped with the SDEJ) should be added to the boot class path if the IBM WebSphere Java SDK is being used.

• `%CURAMSDEJ%\lib\xalan-2.7.2.jar`

• `%CURAMSDEJ%\lib\serializer-2.7.2.jar`

• `%CURAMSDEJ%\lib\xerces-2.9.1.jar`

The boot class path can be set as a JVM parameter. For example:

```
-Xbootclasspath/p: %CURAMSDEJ%\lib\xalan-2.7.2.jar; %CURAMSDEJ%\lib
\serializer-2.7.2.jar;%CURAMSDEJ%\lib\xerces-2.9.1.jar
```

Using a Server Project in Rational Application Developer

The EJBServer server project and CuramSDEJ project needs to be imported and configured for use in Rational Application Developer.

1. Import the project (`.project`) located in the `CuramSDEJ` folder in the development installation.

2. Import the project (`.project`) located in the `EJBServer` folder in the development installation.

Using a Client Project in Rational Application Developer

The webclient client project needs to be imported and configured to run the embedded WebSphere server.

1. Unzip the `%CURAM_DIR%\CURAMCDEJ\doc\RAD\RAD.zip` file into the `%CURAM_DIR%` directory. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\IBM\Curam\Development`. Overwrite the existing files if requested. These default files are for use with Rational Application Developer only.

2. Add the following to the `%CURAM_DIR%\webclient\.classpath` file.

```
<classpathentry kind="con"
path="org.eclipse.jst.server.core.container/
com.ibm.ws.ast.st.runtime.runtimeTarget.v85/was.base.v8">
<attributes>
<attribute name="owner.project.facets" value="jst.web"/>
</attributes>
</classpathentry>
<classpathentry kind="con"
path="org.eclipse.jst.j2ee.internal.web.container"/>
<classpathentry kind="con"
path="org.eclipse.jst.j2ee.internal.module.container"/>
```

3. Add the following to the `%CURAM_DIR%\webclient\.project` file to replace the current buildSpec and natures nodes:

```
<buildSpec>
<buildCommand>
<name>org.eclipse.wst.jsdt.core.javascriptValidator</name>
<arguments>
</arguments>
</buildCommand>
<buildCommand>
<name>org.eclipse.jdt.core.javabuilder</name>
<arguments>
</arguments>
</buildCommand>
<buildCommand>
<name>org.eclipse.wst.common.project.facet.core.builder</name>
<arguments>
</arguments>
</buildCommand>
<buildCommand>
<name>org.eclipse.wst.validation.validationbuilder</name>
<arguments>
</arguments>
</buildCommand>
</buildSpec>
<natures>
<nature>
    org.eclipse.jem.workbench.JavaEMFNature
</nature>
<nature>
    org.eclipse.wst.common.modulecore.ModuleCoreNature
</nature>
<nature>
    org.eclipse.wst.common.project.facet.core.nature
</nature>
<nature>org.eclipse.jdt.core.javanature</nature>
<nature>org.eclipse.wst.jsdt.core.jsNature</nature>
</natures>
```

4. Import the project (`.project`) located in the `CuramCDEJ` folder in the development installation.

5. Import the project (`.project`) located in the `webclient` folder in the development installation.

   - Create a `webclient\work` directory to match the required source directory in the project classpath, if it is not present already.

6. Import the project (`.project`) located in the `CuramEAR` folder in the development installation.

Running the Application in Rational Application Developer

To run IBM Cúram Social Program Management on Rational Application Developer, the relevant servers must be started. Complete the following steps in Rational Application Developer:

1. The server is launched as an RMI Application by executing the StartServer class. From the `EJBServer/ReferencedLibraries/core.jar` (default package), select the `StartServer` class and select the **Run as Application** option from the **Run** button.

2. Enable the server view, by navigating to **Window -> Show View -> Other -> Server -> Servers.**

3. Add CuramEAR to the **WebSphere Application Server** and start the server. To do this right click on the **WebSphere Server v8.5** and click **Add and Remove** Projects. Select **CuramEAR** from **Available Projects** into **Configured Projects** and click **Finish**. Both **WebSphere v8.5 Server** and CuramEAR should now have a status of started and **WebSphere v8.5 Server** should have state synchronized.

4. From the `EJBServer/ReferencedLibraries/core.jar` (default package) in your project, select the `RMILoginClient` class and then select the **Run as Application** option from the **Run** button.

5. The application is now ready for use. Login using the URL : http://localhost:9081/Curam/ AppController.do

Notices

This information was developed for products and services offered in the United States.

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

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

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

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

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

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

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

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

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

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

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Privacy Policy considerations

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

Depending upon the configurations deployed, this Software Offering may use session cookies or other similar technologies that collect each user's name, user name, password, and/or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration and/or other usage tracking and/or functional purposes. These cookies or other similar technologies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

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

Trademarks

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

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

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

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

IBM ®

Part Number: