Trigo Product Center

Release Notes

4.2.0.9a

Note!

Before using this information and the product it supports, read the information in "Notices" at the end of this document.

17 January 2006

This edition of this document applies to Trigo Product Center, version 4.2.0.9a, and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

1. Overview

This document addresses bugs, installation, and notes for the Trigo Product Center 4.2.0.9a on the following platforms:

- WebSphere 4 AE/Oracle 9i

2. Change requests

2.1 Dynamic Attribute Collections have problems in Collaboration Areas

CR: P12888

Issue: Several issues were found with dynamic attribute collections in Collaboration Areas. Note: this issue is related to P13052and P13053.

Solution: Several changes were made to the code to address and correct the issues reported around dynamic attribute collections used in collaboration areas.

2.2 Attribute cache not refreshing correctly

CR: P13052

Issue: Attribute Group cache is not refreshing correctly, either from a change to Specs, Attribute Groups, or from austin.properties setting. Note: this issue is related to P12888 and P13053.

Solution: There were several changes made in the code to address and correct this issue that allows attribute group cache to refresh properly.

Note: This CR includes a required change to the austin.properties configuration file. Please refer to Section 3.2.

2.3 Checkin from Collaboration Area ignores updates to Specs and Attribute Collections

CR: P13053

Issue: Checkin from Collaboration Area ignores updates to Specs and Attribute Collections.Note: this issue is related to P12888 and P13052.

If you add a node to a spec while an item or category is checked out to a collaboration area, and you are using a view that is defined by a dynamic attribute group, then the next time you return to data entry on a workflow step, you will see the new attribute. You can edit it and assign it a value. The item will checkin successfully. However the value you gave to the new spec attribute in the collaboration area is lost, i.e. after checkin the new attribute has a null value in the source catalog or category tree.

Solution: There were several changes made in the code to address this issue. At checkin time, the collaboration will re-determine the set of node Ids that are to be checked in, therefore catching any updates made to specs and attribute collections.

3. Installation Notes

3.1 Changes to austin.properties

After the application of 4.2.0.9a, edit the austin.properties configuration file to include the following configuration information:

```
# Set maximum lifetime of (cache entry for) an attribute group in
#the attribute group cache.
# The numerical value given is interpreted as minutes.
# Setting the value to 0 effectively disables the cache, i.e. an
#attribute group will be fetched from db every time it
# Default value set to five minutes

max_attrgroup_timeout=5
```

Note: If desired, change the value for this setting to increase/decrease the maximum lifetime of an attribute group in the attribute group cache.

3.2 Backup requirements

- Create a full backup of the current Trigo Product Center directories before installation
- It is recommended to apply the new version to a test system to identify any issues before applying to a production system
- Perform a full backup of the database before applying the new version to a production system

Note: Do not delete the old Trigo Product Center version until performing thorough testing with the new installation.

3.3 Removal of J2EE and JSSE

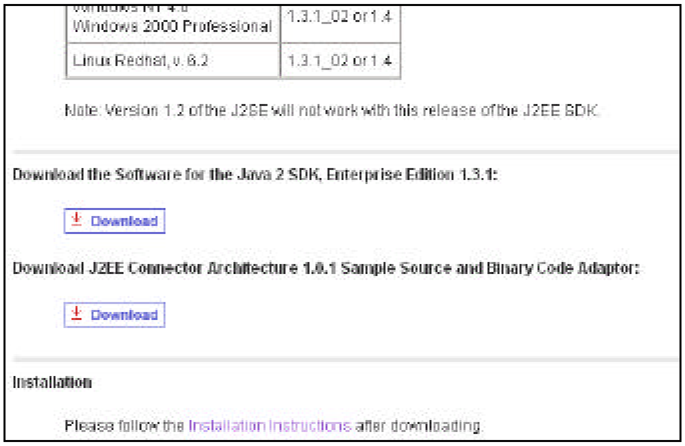
CR: P12188

Due to new company guidelines, J2EE and JSSE are no longer bundled as part of the Trigo installation file; this applies to this patch release 4.2.0.6 and subsequent releases. This is not an issue for customers using WebSphere or WebLogic, only customers using Tomcat.

J2EE - Java 2 Platform, Enterprise Edition

J2EE is a Trigo prerequisite with the use of Tomcat; therefore the J2EE JAR file, which is part of "Version 1.3.1 of the Java 2 SDK, Enterprise Edition ("J2EE" SDK)", will need to be downloaded by the customer from the Sun Microsystems website (**http://java.sun.com/j2ee/sdk_1.3/**). Follow the instructions on the web site to download and install the entire J2EE SDK.

Note: The J2EE jar file is included in the J2EE SDK tar file.



Windows NT 4.0 Windows 2000 Professional	1.3.1_02 or 1.4
Linux Redhat, v. 6.2	1.3.1_02 or 1.4

Note: Version 1.2 of the J2EE will not work with this release of the J2EE SDK.

Download the Software for the Java 2 SDK, Enterprise Edition 1.3.1:

Download J2EE Connector Architecture 1.0.1 Sample Source and Binary Code Adaptor:

Installation

Please follow the Installation Instructions after downloading.

Figure 1 - Link to download J2EE SDK tar file

Once this tar file is downloaded, it is extracted to a location defined by the customer.

Since *j2ee.jar* is no longer included when a Trigo enterprise build is created and it is a prerequisite for Trigo Product Center, the following environment variable needs to be set to identify the location of the *j2ee.jar* file. The environment variable is defined in the *init_trigo_vars.sh* file.

File to edit:

$TOP/setup/init_trigo_vars.sh

Environment variable:

```
J2EE_JAR=(location of j2ee.jar file including the file name)
```

For Example:

```
J2EE_JAR=/opt/j2sdkee1.3.1/lib/j2ee.jar
```

Note: This step is not required for WebSphere or WebLogic.

JSSE – Java Secure Socket Extension

JSSE was being used to provide HTTPS post functionality. After careful review, it was determined that this functionality was not being used, therefore the functionality and the associated JAR file was removed.

4. Installation

Note: This section provides general guidelines on upgrading Trigo Product Center. Some information may differ depending on the methods used for previous installations. Contact your support representative for Trigo Product Center with any installation issues.

4.1 Upgrade installations

This document describes how to upgrade the Trigo Product Center. For this patch release, no database migration is needed for this patch release.

Note: The instructions in this section are for upgrade and patch installations of an existing Trigo Product Center installation running on the same machine. If you are installing Trigo Product Center on a new machine, perform the complete installation steps described in the Trigo Product Center Installation Guide.

4.2 Upgrade preparation

Before attempting to upgrade Trigo Product Center to the latest release version, the following preparation is recommended:

4.2.1 Stopping the whole application on the local machine

1. Check the scheduler to make sure there are no critical jobs that need to be completed. If the queue is clear, kill the scheduler manually by running the script

   ```
   $TOP/bin/go/stop/stop_scheduler.sh
   ```

2. Abort the entire application by running the script:

   ```
   $TOP/bin/go/abort_local.sh
   ```

Every service started on the local machine is aborted. The RMI registry is aborted.

Note: Check to make sure all processes have stopped. Use the following command:
ps –ef | grep ^trigo | grep java

4.2.2 Backup

Before attempting to upgrade Trigo Product Center, it is recommended to backup a copy of the previous Trigo Product Center directories and complete a full backup of the database.

Note: **Do not delete the old Trigo Product Center version until performing thorough testing with the new installation.**

4.3 Installation

4.3.1 Unpack tar file

To extract the installation files, do the following:

Note: GNU tar is needed to untar the Trigo Product Center files.

1. Copy the Trigo Product Center tar file to the user or temporary directory.

```
For example:
{HOME_OF_TRIGO}/tarballs
```

2. Unpack the tar file into the following subdirectory:

   {HOME_OF_TRIGO}/codebases

NOTE: If errors occur during the unpacking process, do not continue with the installation and notify Trigo Product Center support.

Using GNU tar, the following command extracts and unzips the tar file:

Example:

Change directory to the **codebases** directory first.

```
cd /home/trigo/codebases
```

Then tar using an absolute path, for example:

```
tar xfz /home/trigo/tarballs/trigo4216a_websphere_unix.tgz
```

Note: A default subdirectory is created for the new installation. For example, a 4.2.1.6a for a WebSphere/Unix installation creates "`trigo4216a_websphere_unix/`", located in {HOME_OF_TRIGO}/codebases/trigo4216a_websphere_unix.

4.3.2 Change TOP variable

Purpose: To set the shell variable TOP to point to the new Trigo Product Center installation directory.

The settings for the shell environment are defined in the **.bashrc** file.

1. Edit the .bashrc file to point to the new Trigo Product Center directory.
2. Log out and log back in to make sure all new variables are reset with the new parameters.

Note: Enter cd $TOP at the command line to make sure TOP is pointing to the right location. If not, check the .bashrc file for errors.

Example on UNIX:

If the application is installed in the directory:

```
/home/trigo/codebases/trigo4216a_websphere_unix
```

Then, edit the export parameter in .bashrc:

```
export TOP=/home/trigo/codebases/trigo4216a_websphere_unix
```

4.3.3 Verify installation files

Purpose: To verify that all files required by the Trigo Product Center installation are accounted for and have been installed correctly.

Execute the shell script:

```
$TOP/setup/verify_filelist.sh
```

The result is the output of a regular diff command.

- ❖ Lines starting with "<" refer to files added after the installation
- ❖ Lines starting with ">" refer to files missing from the installation

Note: A successful installation shows the following:
Status: Perfect. All the Trigo files are present.

If any files are reported missing, unpack the Trigo Product Center files again and execute verify_filelist.sh again.

4.3.4 Trigo configuration files

This section describes the setup of the configuration files that are needed by the new Trigo installation. Follow the directions as described below for each configuration file.

- ❑ austin.properties
- ❑ admin_properties.xml
- ❑ init_trigo_vars.sh
- ❑ Trigo license file
- ❑ Additional files

❖ **austin.properties**

Desc: On startup, the system will use this file to read in all system level parameters. This file includes settings for the database layer (connection parameters), directory settings, default character sets, thread-pooling parameters, and other settings.

Action: Copy the austin.properties file from the old location into the new location. Then modify the version number field to match the current Trigo version number (i.e. version_number=4.0.1.0e) or comment out this line for the application to generate its own version information.

Old Location: {OLD_HOME_OF_TRIGO}/etc/austin.properties

New Location: $TOP/etc/austin.properties

- **admin_properties.xml**

Desc: This file is used by the administrative utilities to configure clusters of the application.

Action: Copy the admin_properties.xml from the old location to the new location.

Old Location: {OLD_HOME_OF_TRIGO}/etc/admin_properties.xml

New Location: $TOP/etc/admin_properties.xml

- **init_trigo_vars.sh**

Desc: The initialization file is the shell script that initializes the shell variables used by the system.

Action: Copy the init_trigo_vars.sh file from the old location to the new location.

Old Location: {OLD_HOME_OF_TRIGO}/setup/init_trigo_vars.sh

New Location: $TOP/setup/init_trigo_vars.sh

- **Trigo License File**

Desc: The license file contains the license keys needed to use the Trigo application.

Action: Copy the trigo_license.xml file from the old location to the new location.

Old location: {OLD_HOME_OF_TRIGO}/etc/runtime/trigo_license.xml

New Location: $TOP/etc/runtime

❖ **Additional Files**

Files within {OLD_HOME_OF_TRIGO}/etc, that have been configured for the old Trigo environment should be migrated into the new Trigo installation directory.

These may include:

Change of logging levels defined in
{OLD_HOME_OF_TRIGO}/etc/logs/(configfilenames)

Different mount directories for the document store defined in
{OLD_HOME_OF_TRIGO}/etc/docstore_mount.xml

4.3.5 Create/update symbolic link (for Linux/Unix)

Update Link to New Trigo Directory

Update the existing symbolic link to /home/trigo/trigo/. Point to the new Trigo directory.

For example:

Remove the old symbolic link:

```
so rm /home/trigo/trigo
```

Create link to new directory:

```
cd /home/trigo
```

```
ln – s /home/trigo/codebases/trigo4216a trigo
```

Note: This creates /home/trigo/trigo pointing to /home/trigo/codebases/trigo4216a

Create Link to Suppliers Directory

1. To standardize future patch release processes, move the data files from the “suppliers” directory to the “suppliers.orig” directory.

For example:

```
cd {OLD_TRIGO_HOME}/public_html
```

```
mv suppliers ~/suppliers
```

2. Rename the suppliers directory included in the Trigo Installation tarball to “suppliers.orig”.

For example:

```
cd $TOP/public_html
```

```
mv suppliers suppliers.orig
```

3 Then create a symbolic link to the suppliers directory.

For example:

```
cd $TOP/public_html
```

```
ln -s ~/suppliers suppliers
```

Note: All of the work files are to be stored in the above directory. If this link has already been created, update the link to point to the suppliers directory of the new Trigo installation codebases directory.

4.4 Update application server configuration

4.4.1 Tomcat and WebLogic configuration

Action: Copy the server properties file from the old location to the new location.

Depending on the application server used, copy the appropriate properties files.

- For Tomcat installations:
  /etc/tomcat/server.xml
- For WebLogic installations:
  /etc/weblogic/config/trigo_domain/config.xml
- For WebSphere installations

  (configured using WebSphere Administrative Client)

For Tomcat:

Old Tomcat configuration file location: {OLD_HOME_OF_TRIGO}/etc/server.xml

New Tomcat configuration file location: $TOP/etc/server.xml

For WebLogic:

Old WebLogic configuration file location:

{OLD_HOME_OF_TRIGO}/etc/weblogic/
config/trigo_domain/config.xml

New WebLogic configuration file location:
$TOP/etc/weblogic/config/trigo_domain/config.xml

4.4.2 WebSphere configuration

This section provides instructions required to run Trigo Product Center™ with the following WebSphere products:

- IBM WebSphere Advanced Single Server
- IBM WebSphere Advanced Edition

Note: Refer to the "System Requirements" section in Ch. 1 of the Installation Guide for supported versions of WebSphere.

IBM WebSphere Advanced Single Server Configuration

In order to run Trigo Product Center with WebSphere, the following script must be run:

```
$TOP/bin/websphere_install.sh
```

This script handles all of the required WebSphere configuration settings.

Before running the WebSphere script, the following tasks must be completed:

- Trigo tgz file is unpacked
- init_trigo_vars.sh is sourced

Note: Watch the output for errors. If there were no errors, your web archive was installed properly and you can proceed on to section 9. Please do not use the "startServer.sh" script in your WebSphere bin directory to start WebSphere. The Trigo startup scripts will start up WebSphere automatically.

Note: The websphere_install.sh needs to be re-run if any of the port settings in austin.properties has changed

IBM WebSphere Advanced Edition (AE) Configuration

In order to use Trigo Product Center™ 4.0.1 with WebSphere Advanced Edition, follow the configuration instructions outlined in the following sections.

4.4.2.1 Change configuration settings

Start up the WebSphere Administrative Client by running the **`adminclient.sh`** script in your WebSphere bin directory.

After logging into the WebSphere Administrative Console, make the following changes to the configuration:

1. Under Nodes > (your host) >Application Servers > Default Server > JVM Settings Tab, click **Add**.

2. Add the following name/value pairs. The names are case sensitive; therefore, enter the names as they appear below.

Name	**Value**
TOP	Trigo TOP directory
was.home	WebSphere home directory
java.security.policy	Full path to java.policy under TOP
enableJava2Security	true

3. After adding all of the above variables, click **OK**.

4.4.2.2 Using Trigo Scripts with Advanced Edition

- When using WebSphere Advanced Edition, the start_local.sh and stop_local.sh scripts will not shut down WebSphere
- With WebSphere Advanced Edition, start and stop the appsvr module manually using the WebSphere Administrative Console
- After stopping the appsvr module, run the script "abort_appsvr.sh" which is located in $TOP/bin/go/abort. This ensures that the appsvr module has stopped

4.5 Install new WAR file

4.5.1 For Tomcat installations

For added performance, unzip the web archive file for Tomcat into Apache's document root and give the proper permissions for Apache, allowing access to the sub-directories.

Note: If this action has been performed in a previous installation, go to the next section.

Action: Unzip the archive file trigo.war in the old location into the new location.

Old location: {OLD_HOME_OF_TRIGO}/etc/tomcat/webapps/trigo.war

New location: {NEW_HOME_OF_TRIGO}/phtml

Example:

Change directory to phtml

```
cd ~/phtml
```

Unzip the war file into apache's documentRoot directory

```
unzip $TOP/etc/tomcat/webapps/trigo.war
```

Give proper permissions to the files

```
find . -type d -exec chmod +x {} \;
```

4.5.2 For WebSphere installations

- The **`trigo.war`** file is located in the Trigo TOP directory, under $TOP/etc/websphere/webapps/trigo.war
- Install the WAR into WebSphere using the Administrative Console

1. Right-click on “Enterprise Applications”.
2. Click “Install Enterprise Application”.
3. Click “Install stand-alone module (*.war, *.jar)”
4. Enter the path to trigo.war in the “Path” dialog box.
5. Enter an application name (suggested name “trigo”) in the “Application name” dialog box.
6. Enter a context root of “/” in the “Context root for web module” dialog box. Click **Next**.
7. The “Mapping EJB Run As Roles to Users” screen appears. Click **Next**.
8. The “Binding Enterprise Beans to JNDI names” screen appears. Click **Next**.
9. The “Mapping EJB References to Enterprise Beans” screen appears. Click **Next**.
10. The “Mapping Resource References to Resources” screen appears. Click **Next**.
11. The “Specifying the Default Datasource for EJB Modules” screen appears. Click **Next**.
12. The “Specifying Data Sources for Individual CMP Beans” screen appears. Click **Next**.
13. The “Selecting Virtual Hosts for Web Modules” screen appears. Select the virtual host that you would like to run Trigo on and click **Next**.
14. The “Selecting Application Servers” screen appears. Select the application server that you would like to run Trigo on and click **Next**.
15. The “Completing the Application Installation Wizard” screen appears. Click **Finish**.
16. The Trigo WAR file will be installed into WebSphere. After installation, the Trigo application appears under the “Enterprise Applications” tab in the Administrative console. After running the “start_local.sh” script on the Application server, this module can be started by right clicking on it and clicking **Start** from the short menu.

4.6 Upgrading Trigo Product Center database

4.6.1 Schema Changes to support Web Services

If migrating from a version lower than 4.2.0.8 to 4.2.0.8 – 4.2.0.10e, schema changes are required. Run the following Oracle specific SQL script.

```
/*==================================================================
===================*/
/*Script that needs to be run while migrating to 4.2.0.8(or higher
till 4.2.0.10e)     */
/*from versions lower than 4.2.0.8
*/
/*==================================================================
===================*/

/*Create twbs_wbs_webservice table, synonym and constraints */
drop table twbs_wbs_webservice cascade constraints;
create table twbs_wbs_webservice
(
        wbs_id                        number(9, 0) not null,
             wbs_cmp_id                    number(9, 0),
             wbs_name                      varchar2(100) not null,
             wbs_desc                      varchar2(2000),
             wbs_url                       varchar2(2000),
             wbs_wsdl_url                  varchar2(2000),
             wbs_wsdl_doc_path             varchar2(100),
             wbs_protocol                  varchar2(50),
             wbs_impl_script_path          varchar2(2000),
             wbs_store_incoming            char,
             wbs_store_outgoing            char,
             wbs_deployed                  char,
             constraint wbs_0_pk           primary key (wbs_id),
             constraint wbs_1_uk           unique (wbs_name));

drop synonym wbs;
create synonym wbs for twbs_wbs_webservice;

alter table twbs_wbs_webservice drop constraint wbs_0_pk;
alter table twbs_wbs_webservice add constraint wbs_0_pk primary key
(wbs_id);

alter table twbs_wbs_webservice drop constraint wbs_1_uk;
alter table twbs_wbs_webservice add constraint wbs_1_uk unique
(wbs_name);

DROP SEQUENCE qwbs_wbs_id;
CREATE SEQUENCE qwbs_wbs_id START WITH 1 INCREMENT BY 1 NOCACHE
NOCYCLE;


/*Create twbs_wtr_transaction table, synonym and constraints */
drop table twbs_wtr_transaction cascade constraints;
create table twbs_wtr_transaction(
   wtr_id                        number(9, 0) not null,
   wtr_wbs_id                    number(9, 0),
   wtr_cmp_id                    number(9, 0),
   wtr_incoming_path             varchar2(2000),
   wtr_outgoing_path             varchar2(2000),
   wtr_arrival_time              date,
   constraint wtr_0_pk                 primary key (wtr_id));


drop synonym wtr;
create synonym wtr for twbs_wtr_transaction;

alter table twbs_wtr_transaction drop constraint wtr_0_pk;
alter table twbs_wtr_transaction add constraint wtr_0_pk primary key
(wtr_id);

DROP SEQUENCE qwbs_wtr_id;
CREATE SEQUENCE qwbs_wtr_id START WITH 1 INCREMENT BY 1 NOCACHE
NOCYCLE;
```

```
/*==================================================================
==================*/
/*Script that needs to be run while migrating to 4.2.0.11(or above)
from versions     */ /*lower than 4.2.0.8
*/
/*==================================================================
==================*/

/*Create twbs_wbs_webservice table, synonym and constraints */
drop table twbs_wbs_webservice cascade constraints;
create table twbs_wbs_webservice
(
        wbs_id                        number(9, 0) not null,
             wbs_cmp_id                    number(9, 0),
             wbs_name                      varchar2(100) not null,
             wbs_desc                      varchar2(2000),
             wbs_url                       varchar2(2000),
             wbs_wsdl_url                  varchar2(2000),
             wbs_wsdl_doc_path             varchar2(100),
             wbs_protocol                  varchar2(50),
             wbs_style                     varchar2(50),
             wbs_impl_script_path          varchar2(2000),
             wbs_store_incoming            char,
             wbs_store_outgoing            char,
             wbs_deployed                  char,
             constraint wbs_0_pk           primary key (wbs_id),
             constraint wbs_1_uk           unique (wbs_name));

drop synonym wbs;
create synonym wbs for twbs_wbs_webservice;

alter table twbs_wbs_webservice drop constraint wbs_0_pk;
alter table twbs_wbs_webservice add constraint wbs_0_pk primary key
(wbs_id);

alter table twbs_wbs_webservice drop constraint wbs_1_uk;
alter table twbs_wbs_webservice add constraint wbs_1_uk unique
(wbs_name);

DROP SEQUENCE qwbs_wbs_id;
CREATE SEQUENCE qwbs_wbs_id START WITH 1 INCREMENT BY 1 NOCACHE
NOCYCLE;


/*Create twbs_wtr_transaction table, synonym and constraints */
drop table twbs_wtr_transaction cascade constraints;
create table twbs_wtr_transaction(
   wtr_id                          number(9, 0) not null,
   wtr_wbs_id                      number(9, 0),
   wtr_cmp_id                      number(9, 0),
   wtr_incoming_path               varchar2(2000),
   wtr_outgoing_path               varchar2(2000),
   wtr_arrival_time                date,
   constraint wtr_0_pk                   primary key (wtr_id));

drop synonym wtr;
create synonym wtr for twbs_wtr_transaction;

alter table twbs_wtr_transaction drop constraint wtr_0_pk;
alter table twbs_wtr_transaction add constraint wtr_0_pk primary key
(wtr_id);

DROP SEQUENCE qwbs_wtr_id;
CREATE SEQUENCE qwbs_wtr_id START WITH 1 INCREMENT BY 1 NOCACHE
NOCYCLE;
```

```
/*==================================================================
===================*/
/*Script that needs to be run while migrating to 4.2.0.11(or above)
from versions      */
/*4.2.0.8 and higher
*/
/*==================================================================
===================*/
alter table twbs_wbs_webservice add (wbs_style varchar2(50));
```

4.6.2 Using unsupported database versions

Note: If you are using a database version different from the supported version, you will need to copy the database driver JAR file(s) either from the old installation or from the application server lib directory. These JAR files will replace the ones provided in $TOP/jars

For DB2: copy db2java.jar from {OLD_HOME_OF_TRIGO}/jars/db2java.jar

4.7 Test installation

4.7.1 Start Trigo Product Center

To start the Trigo Product Center, execute the following script:

```
$TOP/bin/go/start_local.sh
```

The script starts all the services needed to run Trigo Product Center.

Note: This process should take approximately 30-40 seconds, depending on the speed of the processor.

4.7.2 Check status

Run the **quickrmi.sh** script that was provided by Trigo Product Center and verify the following services have started correctly.

- admin_<machine name>
- appsvr_<machine name>
- eventprocessor
- queuemanager
- scheduler
- workflow

YOU HAVE SUCCESSFULLY INSTALLED TRIGO PRODUCT CENTER!

Notices

IBM may not offer the products, services, or features discussed in this document in all 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.

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 program(s) described in this publication at any time without notice.

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

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

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

IBM Burlingame Laboratory
Director IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
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.

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

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

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

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

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

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

IBM
the IBM logo
AIX
CrossWorlds
DB2
DB2 Universal Database
Domino
Lotus
Lotus Notes

IBM WebSphere Product Center.