



Trigo Product Center, Version 4.2.0.15a Fix Pack Release Notes

Note!

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

28 June2005

This edition of this document applies to Trigo Product Center, version 4.2.0.15a, and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2005. All rights reserved.**

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

CONTENTS

1. ABOUT THIS FIX PACK	2
1.1 New in this release.....	2
1.2 Fix pack requirements.....	2
1.3 Product fix history.....	2
1.3.1 Fix Pack 4.2.0.15a.....	2
1.3.2 Fix Pack 4.2.0.15.....	2
1.3.3 Fix Pack 4.2.0.14.....	3
1.3.4 Fix Pack 4.2.0.12.....	3
1.3.5 Fix Pack 4.2.0.11.....	4
1.3.6 Fix Pack 4.2.0.10b.....	5
1.3.7 Fix Pack 4.2.0.10.....	5
1.3.8 Fix Pack 4.2.0.9a.....	5
1.4 Product compatability.....	5
2. ENHANCEMENTS.....	5
2.1 Web Services Enhancements to support Document/Literal style.....	5
2.1.1 What is contained in this enhancement?.....	5
2.1.2 How the Document/Literal style Web Service works?.....	6
2.1.3 Changes done to support Document/Literal style.....	7
2.1.4 Schema Changes.....	8
2.1.5 Useful links about Document/Literal style of web services.....	11
3. INSTALLATION, MIGRATION, UPGRADE, AND CONFIGURATION INFORMATION.....	11
3.1 Hardware and software requirements.....	11
3.2 Installation notes.....	11
3.2.1 Required update to <i>init_trigo_vars.sh</i>	11
3.2.2 Changes to <i>austin.properties</i>	11
3.2.3 Removal of J2EE and JSSE.....	12
3.3 Preparation.....	13
3.3.1 Stopping the whole application on the local machine.....	13
3.3.2 Backup.....	13
3.3.3 Delete Tomcat working directory.....	14
3.4 Apply Fix Pack.....	14
3.4.1 Unpack tar file.....	14
3.4.2 Verify configuration files.....	14
3.5 Test installation.....	15
3.5.1 Start Trigo Product Center.....	15
3.5.2 Check status.....	16
4. KNOWN LIMITATIONS, PROBLEMS, AND WORKAROUNDS	16
4.1 Namespace must be defined on schema node of WSDL due to DOM versions.....	16
4.2 Enhanced WSDL: cannot change the style.....	19
4.3 <i>getDescendentCategorySetForCategory()</i> always returns empty set.....	19

1. ABOUT THIS FIX PACK

Fix Pack 4.2.0.15a focuses on a series of product fixes that resolve previously reported open defects. The following platform stacks are supported:

- Tomcat Oracle

1.1 New in this release

This release includes the following new and changed features

- Series of product fixes identified in section 1.3.1

1.2 Fix pack requirements

- Fix Pack 4.2.0.15a applies to the base version of Trigo Product Center 4.2.0.9a, therefore Trigo Product Center 4.2.0.9a must be installed prior to the application of Fix Pack 4.2.0.15a.
Note: If working with a version prior to 4.2.0.9a, it is required to upgrade to 4.2.0.9a first and then apply the 4.2.0.15a Fix pack.
- May require update to **init_trigo_vars.sh** (Refer to Section 5.1)

1.3 Product fix history

The change requests listed in this section identify issues that have been reported and resolved with the application of this fix pack.

1.3.1 Fix Pack 4.2.0.15a

Tracking number	Description
17820	Update documentation for checkOutEntries() script operation in the Script Sandbox
JR22007	A user who has NO permission to delete items, because of an ACG, can STILL delete items in the item list view

1.3.2 Fix Pack 4.2.0.15

Tracking number	Description
17241	Rich Search not working
15648	Custom Company Export generates invalid XML
15664	Sequence object settings not exported using exportEnv()
15650	Problems exporting *_CONTENT with exportEnv()
15652	Web Services not exported with exportEnv()
13440	Custom Import/Export Tool - Workflow Import fails for certain workflow types
12612	Custom Import/Export Tool -- trying to Import user settings does not do recursive import of pending attributes
16555, 16578,	Various Import-Export Fixes ported from WPC 5.2 to 4.2.

16648, 16665, 16667, 16600, 16580, 16605, 16601, 16605, 16641, 16693. 16945, 16945, 16949, 17123, 17145, 16997, 17248, 17249	
JR21182, 17487	Script operation "checkOutEntries" invoked at items with attached rule scripts is very slow in execution. Collaboration area corruption by checkoutEntries script op in Entry Preview script. Restricting locales in script poses error in post-processing script.
JR21063, 17578	Item save allowed for non-existent categories
JR21210, 17579	Click on multi-edit item after advanced selection on hierarchy causes exit
JR20976, 16813	Selection Rule Editor displaying Catalog Spec Attributes in an incorrect format, and when used causes application to exit
17850	Clicking on back button on Catalog Script Console causes error message

1.3.3 Fix Pack 4.2.0.14

Tracking number	Description
11313/16275	Save in single-edit item screen seems to fail
14627	If we call useTransaction from the script and there is already a transaction open, it will rollback the open transaction before creating the new one (Refer to section 2.1.1 for more detail)
7578	Changing Name of Category leads to Category Corruption
JR20914	Running delta syndications causes "Unexpected Version" in syndication console, and New or updated items are not exported to masterpack
1563	Distribution Email Input is not saving more than 100 characters
JR20976	Spec mapping GUI in Export Definition displays and generates incorrect attribute path names
16007	Bottom Row Bar hidden in Linked Items Window for screen resolution 768 X 1024
JR21029	Users for role popup window pagination is faulty
JR21095	Undesired full table scans in delete statement on QUE table

1.3.3.1 useTransaction rollback issue

Previously, if we call useTransaction from a script and there is already a transaction open for the DB connection it would rollback the open transaction before creating a new one. Therefore all changes made prior to calling the useTransaction in the script would be lost.

The latest fix commits the previous transactions and opens a new transaction when using useTransaction in the script.

1.3.4 Fix Pack 4.2.0.12

Tracking number	Description
P12480	Hiding the "Versions" column in Lookup Table Console causes the inability to access a lookup table. A web error occurs.
P12482	Access privileges are not working properly. Setting privileges for different roles are not cumulative.
P12491	Not enough character space when entering a distribution addresses. Increase the maximum

	number of characters for the delivery location box in a distribution setup. *Database change required (Refer to Section 2.2.1)
P13049	When a group of attributes from a secondary spec is already added to an attribute collection, and a user removes one of the attributes from the collection some localized attributes for the spec stop appearing in the custom view.
P13494	setAttributeGroupsToProcess does not take account any locales associated to attribute groups.
P13660	When saving multiple attributes in the Multi-edit screen, the save fails.
P13661	For multiple occurrence attributes, it is not possible to add or delete attributes values using the GUI icons until the Save button is clicked. This is a random problem.
P15964	getEntryAttribsList() returns all locale attributes even though some locales have been restricted.
P16003	A user with a role containing locale restrictions destroys catalog data for the role defined restricted locales.
P16032	Changing catalog access privileges from "View Only" to "Editable" gives page not found error.
P16039	After sending an entry and saving the entry, the entry returns to a pre-modified step.

1.3.4.1 Database change required to address P12491

The Max Size of the DST_EMAIL column of the DST table is currently 100 which prevents data of length more than 100 to be saved. The Max Size of this column can be increased to 400 using the following SQL statement:

```
ALTER TABLE TDIS_DST_MKT_DISTRIBUTION MODIFY (DST_EMAIL VARCHAR2 (400))
```

No Bouncing of the application or the database is required. There are no indexes/triggers/contraints defined on the DST_EMAIL column and no other table involves the column. Therefore, it is not required to drop or re-create any indexes.

1.3.5 Fix Pack 4.2.0.11

Tracking Number	Description
P15392	getLinkedItemsForNode broken on multi-valued node.
P15568	getCtgItemAttrib no longer works on multi-occurrence attributes.
P15661	Cannot edit attribute under multi-occurrence group attribute in single edit screen.
P15772	Item lock on updating items through scripting in Web Services Console. The cause was due to the Web Service provider classes not properly cleaning up resources.
P15881	getDisplayValue() returns path if display value is not available
P15962	Lokup table uses Default ACG privileges for deletion
P15963	Hierarchy node search not working
P15999	getLinkedItems() always returns empty list
P16005	Not able to see Items under look up table if not using the Default ACG privileges
P16006	Attribute marked as non-editable is editable in grouping attributes
P16030	getLinkedItemForNode() is not working
P16067	Web Services - Complex data type support. This change request implements Web Services Document Style literal enhancements. Refer to section "3. Enhancements" of this document for details on the Web Services enhancement.

1.3.6 Fix Pack 4.2.0.10b

Tracking Number	Description
P13494	User-defined Core Attribute Collection causes problems with setAttributeGroupsToProcess
P11558	Localized data being deleted in Lookup Tables and Catalogs
P15372	Cannot save an Attribute Collection with locale restrictions more than three times

1.3.7 Fix Pack 4.2.0.10

Tracking Number	Description
P13582	Catalog attribute "ordering" causing "Unexpected Error Occurred" when expanding category tree
P14021	Import Environment getting stuck at 75%
P15117	Issues with "Use Ordering" option
P15214	Datasource of type FTP and FTP functions (getFTP(), sendFTP()) are not working

1.3.8 Fix Pack 4.2.0.9a

Tracking Number	Description
P12888	Dynamic Attribute Collections have problems in Collaboration Areas
P13052	Attribute cache not refreshing correctly
P13053	Checkin from Collaboration Area ignores updates to Specs and Attribute Collections

1.4 Product compatability

Trigo Product Center Fix pack 4.2.0.15a supports the following platforms:

- Tomcat Oracle

2. ENHANCEMENTS

2.1 Web Services Enhancements to support Document/Literal style

2.1.1 What is contained in this enhancement?

This document contains the details of the changes done to the existing web services support in Trigo Product Center. RPC/Encoded style of web services is already available in Trigo Product Center.

However, RPC/Encoded style of web services only supports simple string types, and there is a requirement to support sending and receiving of complex types. This enhancement adds support for Document/Literal style of web services to Trigo Product Center.

In the process of enhancements, existing UI for web services and one of the script operations from earlier version have been modified to work with both styles.

2.1.2 How the Document/Literal style Web Service works?

In order to deploy a Document/Literal style web service, a user would need to create a web service that includes a WSDL defining the schema of the service and a Trigo Product Center trigger script to invoke when a request is encountered. When saving the web service, user would need to explicitly select that it be deployed. Upon deployment, Trigo Product Center will create a URL for the web service where one can access the deployed WSDL. The URL of the web service will take the following form:

`http://<application-webserver>:<application-port-number>/services/<stored-webservice-name>`

Appending the “?wsdl” string to the end of the URL will yield in the path to the stored WSDL for the web service.

A request for a Document/Literal web service would be enclosed in a SOAP envelope, and the body of the SOAP message would include the request document in its entirety. This request document must be in proper XML form, and will be passed to the Trigo Product Center web service handler as-is. A caller would have created this request with prior knowledge of the format of the schema node of the stored WSDL for the web service that is being invoked.

The WPC/TPC web service handling mechanism will receive this request and validate its contents against the WSDL schema for Document/Literal style requests. If the request does not adhere to the WSDL schema, an AxisFault will be thrown. Otherwise, Trigo Product Center will eliminate the namespace references from the request body and pass the modified request to the Trigo Product Center trigger script that was stored at deployment time. (Note: The namespace removal is required due to the limitations of the Trigo Product Center script context’s inability to handle namespace-enabled XML documents.) The Trigo Product Center trigger script will take the contents of the request and use them as defined by the script author. The script must output its results as a valid response to the incoming request. Therefore, the response will be validated against the WSDL prior to returning the output.

Example:

The Document/Literal schema would look like:

```
<element name="getStockQuote" />
  <complexType>
    <sequence>
      <element name="ticker" type="xsd:string" />
    </sequence>
  </complexType>
</element>
<element name="getStockQuoteResponse" />
  <complexType>
    <sequence>
      <element name="response" type="xsd:decimal" />
    </sequence>
  </complexType>
```



```
</element>
```

If the client invoked `getStockQuote("IBM")`, the flow would look like:

1. Trigo Product Center receives a SOAP request from the Axis SOAP stack.
2. Trigo Product Center validates the request message against the above schema.
3. Trigo Product Center strips all namespace prefixes from the request body. Not needed in this case, since this schema defines everything in the default namespace.
4. Trigo Product Center invokes the web service trigger script. The input variables are:

```
- operationName = "getStockQuote"  
- message =  
  "<getStockQuote>  
    <ticker>IBM</ticker>  
  </getStockQuote>"
```

5. The trigger script writes the response to the "out" Writer:

```
- out =  
  "<getStockQuoteResponse>  
    <response>83.76</response>  
  </getStockQuoteResponse>"
```

6. Trigo Product Center validates the response against the above schema.
7. Trigo Product Center sends the entire SOAP response back to the client through the Axis SOAP stack.

2.1.3 Changes done to support Document/Literal style

Following is a list of changes done to support Document/Literal style.

- Web Service Console: A new drop-down list box has been added from where style of web service can be selected. The styles are `RPC_ENCODED` and `DOCUMENT_LITERAL`.
- Database: `wbs` table will have a new column 'style'. This is a varchar column that stores the style of web service.
- There is a new `trigo-axis.jar` that provides the provider class names to axis, named `trigo-axis_1.1.jar`.
- Modification to existing `createWebService` script operation: A new parameter `style` has been added to this script operation that needs to be used to provide the style of the web service. Valid values are `RPC_ENCODED` or `DOCUMENT_LITERAL`.

```
WebService createWebService(String name, String desc, String wsdlDocPath, String protocol,  
String style, String implScriptPath, Boolean storeIncoming, Boolean storeOutgoing, Boolean  
deployed)
```

- Addition of `getStyle()` script operation, which returns the style of the web service as a String.

- New Script Context variables: Two new script context variables are available for scripting - soapOperationName and soapMessage.
 - soapOperationName – This string type variable provides the operation name that was passed as part of the request XML document.
 - soapMessage: This string type variable holds the request message as a XML document.
- For document-literal style web services, the request message sent from the client will be validated against the WSDL stored at web service deployment time. This does NOT affect RPC-encoded style web services.
- Incoming requests for document-literal style web services will be stripped of namespaces when passed to the Trigo Product Center script context (This is due to the fact that the Trigo Product Center script does not support namespace-enabled XML messages).

2.1.4 Schema Changes

Following are the required schema changes for the databases that are being migrated from different versions of the system. These SQL scripts are Oracle specific.

```

/*=====
=====*/
/*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_wsd1_url           varchar2(2000),
    wbs_wsd1_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));
```

2.1.5 Useful links about Document/Literal style of web services

<http://java.sun.com/developer/technicalArticles/xml/jaxrpcpatterns/>

http://searchwebservicestechtarget.com/ateQuestionNResponse/0,289625,sid26_cid494324_tax289201,00.html

3. INSTALLATION, MIGRATION, UPGRADE, AND CONFIGURATION INFORMATION

This section provides general guidelines to apply a Fix Pack to 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.

3.1 Hardware and software requirements

Trigo Product Center 4.2.0.9a, therefore Trigo Product Center 4.2.0.9a must be installed prior to the application of Fix Pack 4.2.0.15a.

Note: If working with a version prior to 4.2.0.9a, it is required to upgrade to 4.2.0.9a first and then apply the 4.2.0.15a Fix pack.

3.2 Installation notes

3.2.1 Required update to *init_trigo_vars.sh*

(This requirement was provided for version 4.2.0.15, please update if this has not been done.)
Update the file **init_trigo_vars.sh** to include the following:

Add the following line after the JDK_HOME variable is set in *init_trigo_vars.sh*:

```
export RMIREGISTRYEXEC_PATH=$JDK_HOME/bin
```

3.2.2 Changes to *austin.properties*

The following configuration change was originally implemented in Fix Pack 4.2.0.9a:

```
# 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.

3.2.3 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 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.

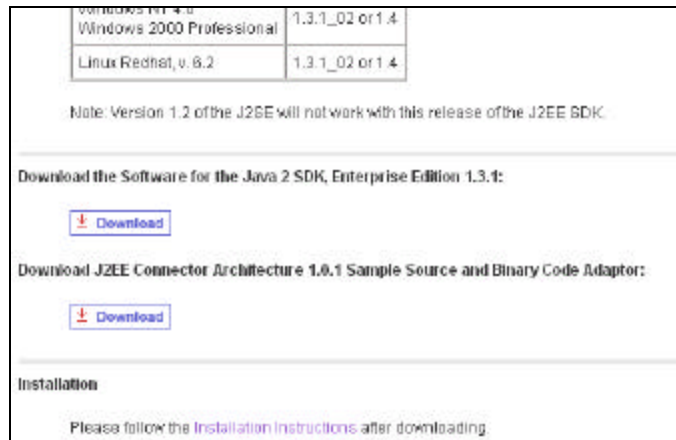


Figure 1 - Link to download J2EE SDK tar file

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:

```
$STOP/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.

3.3 Preparation

Before attempting to apply the latest Fix Pack to Trigo Product Center, the following preparation is recommended:

3.3.1 Stopping the whole application on the local machine

Complete the following steps to stop the Trigo Product Center instance:

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
```

All services running on the local machine is aborted. The RMI registry is aborted.

Note: Check to make sure all processes have stopped using the 'ps' command.

Kill off any rogue "java" or "rmiregistry" processes that remain after shutting down the instance. Occasionally, it may take several attempts to kill off all java processes. Continue killing all java processes until they are all dead.

3.3.2 Backup

- **Create a full backup of the current Trigo Product Center directories before applying the Fix Pack. The Fix Pack will overwrite files that have changed. If any issues occur, the backup will allow a rollback to a previous version**
- **It is recommended to apply the Fix Pack to a test system to identify any issues before applying the Fix Pack to a production system**
- **Perform a full backup of the database before applying the Fix Pack to a production system**

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

3.3.3 Delete Tomcat working directory

For configurations using Tomcat, delete the Tomcat working directory using the following command:

```
rm -rf $TOP/etc/default/tomcat33/webapps/trigo
```

Once the working directory has been deleted, restart the application server and apply the Fix Pack.

3.4 Apply Fix Pack

To apply the Fix Pack to Trigo Product Center, complete the following tasks:

- Unpack tar file
- Update configuration files
- Test installation

3.4.1 Unpack tar file

Purpose: To extract and update any new installation files into the current working directory

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.

Example:

```
{HOME_OF_TRIGO}/tarballs
```

2. CD to \$TOP, the current working directory, and unpack the tar file:

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

Example:

Tar using an absolute path:

```
tar zxvf /home/trigo/4_1/trigo_4107a_05_fixpak_tom_ora.tgz
```

3.4.2 Verify configuration files

Verify all configuration files required by the new installation and make any updates as needed. Refer to the backup copy of the configuration files for the previous installation if needed.

- austin.properties

- ❑ admin_properties.xml
- ❑ init_trigo_vars.sh
- ❑ Trigo license file

❖ **austin.properties**

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, which are documented in the file.

File location: \$STOP/etc

❖ **admin_properties.xml**

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

File location: \$STOP/etc

❖ **init_trigo_vars.sh**

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

File location: \$STOP/setup

❖ **Trigo License File**

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

File location: \$STOP/etc/runtime

3.5 Test installation

3.5.1 Start Trigo Product Center

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

```
$STOP/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.

3.5.2 Check status

Run the **STOP/bin/go/rmi_status.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 APPLIED THE LATEST FIX PACK FOR TRIGO PRODUCT CENTER!

4. KNOWN LIMITATIONS, PROBLEMS, AND WORKAROUNDS

This section identifies known issues for this release and provides a workaround, if one is available. These known issues will be addressed in a future Fix Pack.

4.1 Namespace must be defined on schema node of WSDL due to DOM versions

Due to limitations caused by the XML parsing implementation shipped with Trigo Product Center 4.2.0 (provided by Xerces version 2.4.0), the namespace declaration must be defined locally on the schema node of the WSDL. This will be noticed mostly when deploying Document-Literal style web services. For example, the following is a valid WSDL, which would not be correctly recognized by Trigo Product Center:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:y="http://ibm.com/wpc/test/stockQuote"
targetNamespace="http://ibm.com/wpc/test/stockQuote">
  <types>
    <xs:schema
targetNamespace="http://ibm.com/wpc/test/stockQuote"
elementFormDefault="qualified">
      <xs:element name="getStockQuote">
        <xs:complexType>
          <xs:sequence>
```

```

                <xs:element name="ticker"
type="xs:string" nillable="false"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="getStockQuoteResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="response"
type="xs:decimal"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
</types>
<message name="getStockQuoteRequest">
    <part name="parameters" element="y:getStockQuote"/>
</message>
<message name="getStockQuoteResponse">
    <part name="parameters" element="y:getStockQuoteResponse"/>
</message>
<portType name="StockQuotePortType">
    <operation name="getStockQuote">
        <input message="y:getStockQuoteRequest"/>
        <output message="y:getStockQuoteResponse"/>
    </operation>
</portType>
<binding name="StockQuoteBinding" type="y:StockQuotePortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getStockQuote">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="StockQuoteService">
    <port name="StockQuotePort" binding="y:StockQuoteBinding">
        <soap:address
location="http://localhost/axis/services/StockQuoteService"/>
    </port>
</service>
</definitions>

```

The WSDL would have to be written, as follows, to be correctly parsed:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:y="http://ibm.com/wpc/test/stockQuote"
targetNamespace="http://ibm.com/wpc/test/stockQuote">

```

```

<types>
  <xs:schema
targetNamespace="http://ibm.com/wpc/test/stockQuote"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <xs:element name="getStockQuote">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ticker"
type="xs:string" nillable="false"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="getStockQuoteResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="response"
type="xs:decimal"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</types>
<message name="getStockQuoteRequest">
  <part name="parameters" element="y:getStockQuote"/>
</message>
<message name="getStockQuoteResponse">
  <part name="parameters" element="y:getStockQuoteResponse"/>
</message>
<portType name="StockQuotePortType">
  <operation name="getStockQuote">
    <input message="y:getStockQuoteRequest"/>
    <output message="y:getStockQuoteResponse"/>
  </operation>
</portType>
<binding name="StockQuoteBinding" type="y:StockQuotePortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getStockQuote">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="StockQuoteService">
  <port name="StockQuotePort" binding="y:StockQuoteBinding">
    <soap:address
location="http://localhost/axis/services/StockQuoteService"/>
  </port>
</service>
</definitions>

```

4.2 Enhanced WSDL: cannot change the style

Case ID: P16059

Create a web service using DOCUMENT-LITERAL. Save and go back to the newly created web service and change the style to RPC-ENCODED and save again. The style DOCUMENT-LITERAL is displayed.

This is a known limitation. A user cannot change the style of a web service that has been deployed.

4.3 getDescendentCategorySetForCategory() always returns empty set

Case ID: 17246

When using the script function getDescendentCategorySetForCategory(), it always returns an empty set This issue only exists in environments using DB2.

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

MQIntegrator
MQSeries
Tivoli
WebSphere

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

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

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

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

IBM WebSphere Product Center.