Release Notes for WebSphere Product Center Fix Pack 5.0.1.11

Note!

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

20 December2005

This edition of this document applies to WebSphere Product Center, version 5.0.1.11, and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

1. ABOUT THIS RELEASE

This document addresses enhancements, product fixes, and installation instructions for WebSphere Product Center Fix Pack 5.0.1.11

Fix Pack requirements

- WebSphere Product Center 5.0.1 must be installed prior to the application of Fix Pack 5.0.1.11

- **CONFIGURATION AND/OR SCHEMA CHANGES MAY BE REQUIRED WITH THE APPLICATION OF THIS FIX PACK.**

  REFER TO "SECTION 3.1- INSTALLATION NOTES" FOR MORE INFORMATION

1.1 New in this release

The major focus for 5.0.1.11 was to provide:

- Implementation of Java API
- Resolution to previously reported issues

2. PRODUCT FIX HISTORY

2.1 Fix Pack 5.0.1.11

2.1.1 Functional Enhancements

2.1.1.1 Java API implementation

The Java API and Java WebServices capabilities, implemented as part of this maintenance release, are two distinct layers of WPC capability. The base layer, the Java API, provides an Object Oriented mechanism to expose WPC artifacts, whilst the Java WebServices support provides the capability to develop and deploy java based WebServices written using the Java API.

Note: For this release only the java WebServices environment supports the use of the Java API.

Refer to Section 5 for detailed information on the Java API implementation.

2.1.2 Completed change requests

Tracking Number	Description
19351	NullPointerException when editing item through a catalog, but not in a workflow.

2.2 Fix Pack 5.0.1.10

2.2.1 Functional enhancements

2.2.1.1 Java API and Java Web Services Support

Java API and Java Web Services support have been implemented in this Fix Pack. For details information, refer to Section 5 of this document.

2.2.2 Completed change requests

Tracking Number	Description
14088	Import/Export Tool - Spec Node of type Lookup Table is not being imported correctly
17779, 17780, 17781	Import/Export Tool - Exporting the Lookup Table Content produces a blank CSV file.
17783	Import/Export Tool - Unable to export out Web Services object
19276	Use default soap envelope schema in case property is not defined in common.properties
05022,147,000/ JR22019	SOAP Envelope URL is inaccessible from behind a firewall. Needs to be user configurable by adding it to common.properties
05029,147,000/ JR22102	SOAP Envelope does not get stripped correctly
05027,147,000	getSourceEntrySetRelatedEntries() does not return a value
05030,147,000/ JR22110	Hierarchy pre-processing script and the post processing script run twice
90122,001,866/ JR22033	Incorrect handling of empty but required item attribute values in workflow step after second save
90122,001,866/ JR22034	No error messages shown for required multi-occurrence lookup attribute values in a workflow step if value is left as none
90506,001,866. JR22064	Required attribute within a multi-occurrence group bug (In a workflow step if an attribute belonging to a grouped attribute is set to required, the grouped attribute is also set required).
79704,019,866/ JR22087	Collaboration area explorer in left navigation pane shows incorrectly a large number of results on DB2 (not on Oracle).
84698,001,866/ JR21990	Data entry screen no longer checks for string length.
84066,001,866/ JR21845	Classcaseexception is thrown when executing savemultipartrequestdata
03099,49R,000/ JR21820	OUTOFMEMORY error on parseDelim script operation when the delimiter is an empty string

2.3 Fix Pack 5.0.1.9

2.3.1 Functional enhancements

2.3.1.1 Script operation changes

A new optional boolean parameter has been added to the createWebService() operation. If set to true, then authentication will be required to invoke this web service:

- createWebService(name, desc, wsdl, protocol, style, scriptPath, storeIn, storeOut, deployed, **[authRequired]**)

The corresponding methods are available on the WebService script object to access the new parameter:

- void WebService::setAuthRequired(boolean authRequired)
- boolean WebService::isAuthRequired()

In order to support calling into a web service that has authorization a modified version of the invokeSoapServer() method which specifies a username and password:

- invokeSoapServer(URL, method, values, names, username, password)

The username must be of the form "user@company". If authentication is required, then the authentication must be done using a username in this format. If the credentials in the request do not match a valid WebSphere Product Center user, then the request will fail with a Server Unauthorized fault code and HTTP error 401. If no username is specified on a web service for which authentication is required or no company has been specified then a Server, an unauthenticated fault code is returned, also with a HTTP error 401.

2.3.1.2 Adding user JARs

Within the enterprise install dir, edit the file `$TOP/etc/default/user_jars.list` and add the jar to the file, as a line specifying the full path to the jar. Use of environment variables within the path is allowed as they will be resolved when `$TOP/setup/init_ccd_vars.sh` is run.

Stop WebSphere Product Center, re-exec `$TOP/setup/init_ccd_vars.sh`, there will be a message saying how many user supplied JARs have been added to the environment.

For WebSphere Application Server only:

Run `$TOP/bin/websphere/refresh_war.sh` to update the WebSphere Product Center web application with the new user jars. After updating, use `$TOP/bin/websphere/install_war.sh` to redeploy the updated WebSphere Product Center web application to WebSphere Application Server.

Start WebSphere Product Center, and you will be able to use classes from the user JARs within script, without requiring a loadJar statement for them.

2.3.2 User interface change

2.3.2.1 Update to Web Service Console user interface

The user interface for the Web Services Console has been modified to include a checkbox that indicates if the web service requires authorization. This corresponds to the authRequired parameter on the createWebService() call.

2.3.2.2 Workaround for Missing the "Define Location Specific attributes" button on the catalog attributes page

Workaround for the missing the "Define Location Specific attributes" button on the catalog attributes page.

1. To add location specific data to a catalog

void Catalog::defineLocationSpecificData(CategoryTree ctr, Spec spc, AttrGroup[] inhAttrGrps)

2. For removing location specific data

void Catalog::removeLocationSpecificData(CategoryTree ctr)

2.3.3 Completed change requests

Tracking Number (PMR/APAR)	**Description**
05012,147,000 JR21932	Unable to pass a date to Web Service

2.4 Fix Pack 5.0.1.8

- Functional enhancements to allow JDBC access using script operations and the ability to drive JAVA methods and constructors from WPC scripts via reflection

- A number of changes to optimize performance

- User Interface changes

- APAR fixes to customer reported PMRs

2.4.1 Functional enhancements

2.4.1.1 Scripting operations for JDBC access

getWPCDBContext()

Return the DBContext object which manages and controls all the WPC database coordination. This function wraps standard AustinContext methods

- Prototype: AustinContext::getWPCDBContext()
- Input: None, uses the global AustinContext
- Returns: a DBContext object
- Description: Get the database context object

getWPCDBConnection()

Provides a connection to the WPC database using the database context object. This function wraps standard DBContext methods.

- Prototype: Connection DBContext::getWPCDBConnection()
- Input: A DBContext Object
- Returns: A Sql Connection
- Description: Get a connection using the DB context

releaseWPCDBConnection()

Provides the correct operations for releasing a connection to the WPC Database. This function wraps standard DBContext methods.

- Prototype: Connection DBContext::getWPCDBConnection()
- Input: A Sql Connection, DBContext
- Returns: void
- Description: Release a connection using the DB context

openJDBCConnection()

Provides functionality for accessing a database using JDBC. This function wraps a standard JDBC method

- Prototype: Connection JDBCConnector::openJDBCConnection()
- Input:
  Sring: Driver name (e.g. "COM.ibm.jdbc.db2.app.DB2Driver")
  String: A database URL (e.g. "jdbc:db2:myDatbase")
  String: A userid
  String: Database password
- Returns: A Sql Connection
- Description: Get a Connection using JDBC Drivers

releaseJDBCConnection()

Provides the correct operations for releasing a connection retrieved by JDBC. This function wraps a standard JDBC method

- Prototype: Connection JDBCConnector::releaseJDBCConnection()
- Input: A Sql Connection
- Returns: void
- Description: Commit a transaction using the DB Connection

commit()

Wraps the standard java.sql.Connection.commit functionality. Note that the wrapped operation always turns the auto commit function back on. Consequently this wrapper ensures it is turned off again after committing. Used in conjunction with the executeBatchUpdate() function.

- Prototype: Connection::commit()
- Input:	A Sql Connection
- Returns: void
- Description: Commit a transaction using the DB Connection

rollback()

Performs standard java.sql.Connection.rollback functionality. This is a script wrapper around standard java Connection functionality.

- Prototype: Connection::rollback()
- Input:	A Sql Connection
- Returns: void
- Description: Rollback a transaction using the DB Connection

executeQuery()

Executes a standard executeQuery method on a Connection byr creating a statement. This is a script wrapper around standard java Connection functionality.

- Prototype: Connection::executeQuery()
- Input:	A Sql Connection
- Returns: ResultSet
- Description: Execute the query using the Connection object, returning the ResultSet.

executeUpdate()

Executes a standard executeUpdate method on a Connection after creating a statement. This is a script wrapper around standard java Connection and prepared statement functionality.

- Prototype: Connection::executeUpdate()
- Input:	A Sql Connection
- Returns: int – number of rows affected
- Description: Execute the query using the Connection object. Returns the number of rows inserted, deleted or updated.

executeBatchUpdate()

Executes a batch update and should be used in conjunction with the commit and rollback functions. This is a script wrapper around standard java Connection and prepared statement functionality.

- Prototype: Connection::executeBatchUpdate()
- Input: A Sql Connection, String sql, Object[][] batchValues
- Returns: void
- Description: Executes a prepared statement for a batch update using the Connection object. Returns the number of rows inserted, deleted or updated. The Object[][] is a HashMap of HashMaps, each indexed by integer, whose value is the replacement for a '?' in the prepared statement for a given batch.

next()

Wraps the iterator function provided by the Java ResultSet object. This is a script wrapper around standard java ResultSet functionality.

- Prototype: resultSet::next()
- Input: A ResultSet
- Returns: Boolean true or false row returned
- Description: Move the ResultSet iterator to the next result. Returns false if it has iterated past the last result.

getColumn()

Retrieves the column value in a result set row, based upon column name. This is a script wrapper around standard java ResultSet functionality.

- Prototype: resultSet::getColumn(String colName)
- Input: A ResultSet, String, the column name
- Returns: Object
- Description: Get the entry for the current result at column colName. Returns an object of type Integer, String, or Date (depending on the data type of the column).

getColumnAt()

Retrieves the column value in a result set row, based upon column position. This is a script wrapper around standard java ResultSet functionality.

- Prototype: resultSet::getColumnAt(Number position)
- Input: A ResultSet, Int the column position
- Returns: Object
- Description: Get the entry for the current result at a column position. Returns an object of type Integer, String, or Date (depending on the data type of the column).

loadJar(String filename)

To access the JDBC drivers specifically required to be used by the openJDBCConnection function it is required to add the containing JAR file dynamically.

- Prototype: Boolean Connectort::loadJar(String filename)
- Input: void
- Returns: Boolean.TRUE is load successful otherwise Boolean.FALSE
- Description: Appends a Jar file to the runtime classpath. The name of the Jar is passed as the argument, which describes the absolute file path on the server.

2.4.1.2 Java Reflections

Java Reflections was provided to introduce the ability to drive JAVA methods and constructors from WPC script via reflection. The introduction of Java Reflections allows the following:

- Move complex scripts into Java libraries (which would be easier to develop and debug)
- Leverage existing Java libraries (for instance for integration with other applications) immediately without waiting for a new version or patch of the product.

The following script calls are provided as changes to external or Key APIs:

1) Constructor createJavaConstructor(String className, [String type0 , String type1……….type9]

This creates a java.lang.reflect.Constructor Object using the supplied className and types.

2) Object runJavaConstructor(Constructor c, [Object arg0 , Object arg1……….arg9]

This runs the Constructor c and creates a new instance of the Object using the supplied args.

3) Method createJavaMethod(String className,String methodName, [String type0 , String type1……….type9]

This creates a java.lang.reflect.Method Object using the supplied className methodName and arguments.

4) Object runJavaMethod(Method m, [Object arg0 , Object arg1……….arg9]

This invokes the Method m using the supplied arguments. The object returned is the Object that the invoked Method returned.

5) Object createJavaArray(String typeName, Integer dim0 [Integer dim1......Integer dim9]

This creates a java array of type typeName. The number of supplied integers indicates the number of required dimensions.

The internals include a new reflect.xml script operations file that exposes the static script calls. It calls through to static methods in ConstructorMirror and MethodMirror, which perform the checking of the supplied parameters and runs the request.

Reflect.xml

- This is the module that provides the definition of the script call. The optional arguments are passed parsed into an array and the java calls are issues to the classes in common.reflect.

Mirror.java

- This class holds all of the common logic for the constructor and Method, this includes the exception message which is appended to as we process a request so when an Exception is issued the message contains the context of the failure.
- It contains the logError method, which logs any Exceptions.
- It validates the classname passed in.

ConstructorMirror

- This exposes the create and run Constructor methods for the script to use.
- Constructors check that an interface is not supplied as a classname.
- No checking is performed on the arguments passed in on the runJavaConstructor – as java will perform this checking.

MethodMirror

- This exposes the create and run Method methods for the script to use.
- No checking is performed on the arguments passed in on the runJavaMethod – as java will perform this checking

ReflectionUtils

Two static utility methods are introduced:

1) A method to convert types to Classes this is called by the mirror class when createJavaMethod and Constructor need their parameter types converted to Classes.

2) A method to expose the creation of Arrays. This is called directly by the java code for the createJavaArray operation in the reflect.xml file.

2.4.2 User interface changes

2.4.2.1 Collaboration Area Explorer

The Collaboration Area Explorer allows a user to quickly browse a collaboration area and its items. For each collaboration area, the user can add a module for it in the left pane. At the top level, the Collaboration Area Explorer lists the collaboration area's steps and the number of items in each step (if any). A context menu allows the user to quickly jump to the definition of the workflow used by this collaboration area.

Adding a Collaboration Area Explorer module to the left pane

In the left pane drop down, a new selection has been added for Collaboration Areas. This new section will list the names of all the collaboration areas the user has access to and should correspond to the same list the user sees in the collaboration area console. Selecting an entry in this list will create a corresponding Collaboration Area Explorer module. Only one Collaboration Area Explorer module instance is allowed per Collaboration Area.

2.4.2.2 Features

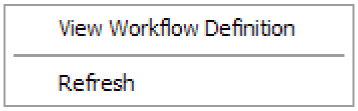
1. All steps in the collaboration area are listed in alphabetic order. A page will be displayed if more than 10 steps are present in the collaboration area.
2. Next to each step name is a count of the number of items currently in that step. The count is omitted if there are no items in the step.
3. A green refresh button is provided to refresh the contents of the collaboration area. Using this refresh button is equivalent to removing the module from the left pane and adding it again.
4. When a step is expanded, the items in the steps are displayed. If the step contains more than 10 items, they are paged. Items that are reserved will have the icon (▣) to indicate that the current user reserved the item, and the icon (▣) to indicate some other user reserved the item. These icons will also have alt-text to indicate this.

Handling Collaboration Area Structural Changes

If after adding a collaboration area to the left pane, the underlying collaboration area is deleted, or it’s workflow has steps removed, an error message should be displayed to the user and the explorer should be removed from the left pane.

Context Menus

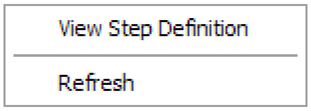
Explorer Context Menu



This menu appears by right clicking in any area of the Collaboration Area Explorer where a step context menu or an item context menu is not active (see following sections). Currently the menu has two entries:

- View Workflow Definition - Selecting this entry will bring up in the right pane a read-only view of the workflow used by this collaboration area.
- Refresh - This will refresh the whole Collaboration Area Explorer module. The end result should be functionally equivalent to removing the module from the left pane and adding it again.

Step Context Menu



This menu appears by right clicking on the text of any step of Collaboration Area Explorer. Currently the menu has two entries:

- View Step Definition - Selecting this entry will bring up in the right pane a read-only view of step in context.
- Refresh - This will refresh the step in context. The item count will be updated. If the step is expanded refresh will leave the step expanded to the same page. If the page is invalid because items are no longer in the step the last valid page will be displayed.

Item Context Menu

This menu appears by right clicking on the text of any item of Collaboration Area Explorer. Currently the menu has two entries:

- Reserve - Selecting this entry will reserve the item for the user and update the item's icon to indicate this. This entry should be enabled if the equivalent button in the widget screen is enabled. If the reserve fails, a popup error message will be displayed.
- Release - Selecting this entry will release the item and update the item's icon to indicate this. This entry should be enabled if the equivalent button in the widget screen is enabled. If the release fails, a popup error message will be displayed

2.4.2.3 User interface change - Icon to distinguish items with checked out attributes

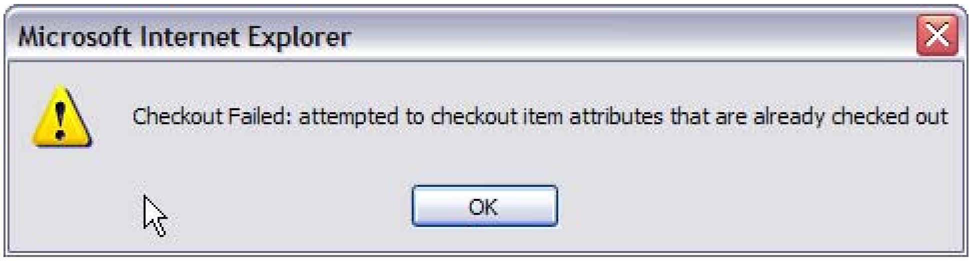
Items that have been checked out to a collaboration area are now displayed with a ( ) icon. The icon also has an alt-text that indicates this. Note that this feature does not provide information about what attributes are checked out to which collaboration area.

This feature is available for items in a category and for item search results.

New entry "Checkout" in the context menu for Items

A new "Checkout" entry has been added to the context menu for Items. This entry allows a user to checkout an item to a collaboration area directly from the left pane. The list of collaboration areas in this menu should be the same as the corresponding list found in the widget UI, if the item is brought up there.

Checking out an item via this menu will bring up a modal progress dialog that is closed automatically if the checkout is successful. Upon successful completion of the operation, the icon for the item will be updated to reflect the item is checked out. If an error occurs or the checkout fails, a popup message will be shown to the user:


Microsoft Internet Explorer

Checkout Failed: attempted to checkout item attributes that are already checked out

2.4.2.4 *Behavior changes*

The 'Define Location Attributes' button will no longer be present in the catalog attributes screen if the catalog contains items. This would prohibit the deletion of the location hierarchies that are already setup for that catalog.

2.4.3 Completed change requests

Tracking Number (PMR/APAR)	Description
28947,999,000 JR21452	New items not showing up in step when importing into collaboration area
19905,999,000 JR21395	Collaboration area item does not appear in any step but displays as being checked out
27172,999,000 JR21449	Catalog Rollback - DB2 constraint on item table
28795,999,000 JR21462	deleteEntryNode() method does not remove the entry node for multi-occurrence attributes
43858,550,000 JR21426	Checkouts not working for categories
43391,550,000 JR21355	Lookup table does not return the primary key when used in a workflow
43505,550,000 JR21354	lkpTable.getLkpKeys() produces an error
43859,550,000 JR21420	Changing a spec causes runSearch to fail
17985	Calling dropEntry() immediately after checkoutEntry() leaves ghost items in workflow. To remove the ghost entries that are already present in the existing database run the following script: $TOP/src/db/schema/cmp/create_cmp.sh. The script will delete the existing company, if one exists, and create a new company. **Note: Create a full backup of the current WebSphere Product Center directories before deleting and creating a new company.**

2.5 Fix Pack 5.0.1.7

2.5.1 Performance improvements

Item	Description
Import Optimization	The following changes were made to improve the performance for imports: • EntryNode objects are not tracked during imports • The import code does not retrieve the item again if that same item was already retrieved in the script
Item Save Optimization	The following changes were made to improve the performance for item saves: • The serialized form of an item and category is saved as a BLOB in one row. Due to this change the schema will need to be re-created • The 'Cheney Algorithm' is now being used for garbage collection • Grouping nodes with a min and a max occurrence of 1 are no longer validated during item save • Only the changed attributes are merged when saving an item in a workflow with split steps • The algorithm for expanding and collapsing empty available locations was optimized

2.5.2 Completed change requests

Tracking Number (PMR/APAR)	Description
05355,999,000/ JR21164	Workflow items are made persistent in Collaboration Area
04246,999,000/ JR21149	The item in a collaboration area of a small workflow is freezing in a merge step
08462,999,000 /NA	Imports running out of memory

2.6 Fix Pack 5.0.1.6

2.6.1 Enhancements

Validation Optimization

Optimizing the validation policies in WebSphere Product Center has improved performance in the Item Save from Location Attributes popup and Item Save from Single Edit screen.

Old Validation Policy

- Always validate all attributes regardless of any circumstances on every save
- Results in extremely large numbers of unchanged attributes validated at every save

- Items in workflows can be "saved as draft", i.e., with validation errors, and the validation errors are caught again on the next save

New Validation Policy

- Validation occurs on any changed attribute
- Min/max occurrence validation is performed on unchanged required attributes. If this was not the case, new items would pass validation without required data
- Attributes with previous validation errors, from "Save as Draft", are validated

New Validation Policy Omissions

- If an item is saved in the database, and then one of its specs has a validation rule changed (other than min occurrence), this validation rule will never be applied to this item unless the relevant attribute is edited. For example, if a string spec attribute gets a shorter max length, that string attribute is only validated if the attribute changes.
- If a validation rule for one attribute depends on another attribute, such as "retail price is greater than wholesale price", and the wholesale price changes, a validation on retail price will not happen. The workaround for this issue includes putting the validation rule on both wholesale and retail price, or using a script for validation.

2.6.2 Performance improvements

Item	Description
Validating only changed attributes	When an item was being saved, the majority of the time was being spent in validating all the global and location attributes (changed or unchanged). The behavior was modified so that only the attributes that are changed are validated. More information can be found in Section 2.1.1 Validation Optimization.
Not populating default values for location attributes	Populating the location attributes with default values, sequences and value rules was taking a significant portion of time when saving an item. To address this problem a configuration parameter was added in the file "common.properties" that should be set to false if there are no default values, sequences or value rules on location attributes in order to improve the time taken to save an item. Refer to section 4.1.1 Configuration Changes.
Optimizing non-split workflow steps	A large portion of time it took to save an item in a workflow step was being spent in copying all the attributes of the item to cover the case of splitting the item in multiple steps. This was improved to only perform the copying of attributes when there is a split in workflow steps as oppose to always perform the operation.

2.6.3 Completed change requests

Tracking Number	Description
JR20869	Cannot set a value for location attributes of type "Lookup Table"
JR20876	Cannot get differences between a checkin clone entry and an original container entry
JR20920	Runjob executable always returning 0 for the return code
JR20982	Lookup table popup is not working from the location screen
JR20988	User locale value is not accessible in workflow step scripts

JR20988	Company code value is not accessible in workflow step scripts
JR21003	Setting an invalid value for a lookup table attribute throws a Null Pointer Exception
JR21044	If a spec is modified, the changes are not reflected on any existing items using the spec
16700	Inconsistent behavior between global and location for lookup table attributes
16701	Workflow table is returning duplicate rows
16769	stepPath is null in the postSave script when saving the step entry from an import
16770	getOriginalEntry is returning the same collaboration area entry in the postSave script of the checkin operation
16939	Query Catalog Activity gives server error page

2.7 Fix Pack 5.0.1.5

The main purpose of this fix pack is to 'foreport' extensions to Web Services functionality made and delivered in in a previous release.

2.7.1 Enhancements

Web Services Enhancements to support Document/Literal style

What is contained in this enhancement?

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

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

How the Document/Literal style Web Service works in Websphere Product Center?

In order to deploy a Document/Literal style web service, a user would need to create a web service, which includes a WSDL defining the schema of the service and a Websphere Product Center trigger script to invoke when a request is encountered. When saving the web service, the user would need to explicitly select that it be deployed. Upon deployment, Websphere 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 Websphere 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, which is being invoked.

The Websphere Product Center 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, Websphere Product Center will eliminate the namespace references from the request body and pass the modified request to the Websphere Product Center trigger script, which was stored at deployment time. The namespace removal is required due to the limitations of the Websphere Product Center script context's inability to handle namespace-enabled XML documents. The Websphere 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. Websphere Product Center receives a SOAP request from the Axis SOAP stack.

2. Websphere Product Center validates the request message against the above schema.

3. Websphere 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. Websphere 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. Websphere Product Center validates the response against the above schema

7. Websphere Product Center sends the entire SOAP response back to the client through the Axis SOAP stack.

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 ccd-axis.jar that provides the provider class names to axis, named *ccd-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 Websphere Product Center script context (This is

due to the fact that the Websphere Product Center script does not support namespace-enabled XML messages).

2.7.2 Completed change requests

Tracking Number	Description
16336	Add support for Document-Literal style web services to the current web service functionality. **Refer to section 2.1.1 for more information.**
16337	Item lock when updating item through script in Web Services Console
16338	WSDL string styles should be consistent for both script operation and UI
16339	A fault should be returned when a script has a runtime error
16342	The soap fault code should default to "soapenv:Server" if not specified
16677	Invalid WSDL produces error and gets saved

2.8 Fix Pack 5.0.1.4

The major focus for 5.0.1.4 was to provide a number of performance improvements to improve end user experience, item retrieval, and save performance. Additionally, several issues have been resolved with the change requests that have been completed for this release.

2.8.1 Performance improvements

Item	Description
Excessive calls to syncToModel	The serialized form was being synchronized repeatedly with Specs even though no changes were made to the Specs. Synchronization now occurs only for situations when the meta-data model changes.
Implement Lazy initialization of caches	System information about location data is currently cached when an Item is retrieved and only used when operations on location data are run. Loading of the cache has been made a lazy operation and only takes the hit when needed.
Lookup Table Cache DB Calls	Loading a lookup table from the cache to resolve attribute values for Nodes of type Lookup table runs a query to determine a timestamp each time. Caching has been modified to run the query only once during the processing of a task; hence the query will not run more than once during the rendering of a UI screen or running of an export.
Multiple retrieval of Items for Workflow screen	Item data is loaded multiple times when a user accesses the workflow multi-edit screen. Item data is not reloaded unless needed.
Low level optimizations	Multiple low-level optimizations to Java code were made to improve performance and memory usage.
Interning of Paths to EntryPaths	Attribute paths for data are represented as strings in scripts and are interned on each call to get or set an attribute. Interned paths have been cached for faster access.
Rendering Asynchronous UI Events	A sort is performed on the list of UI widgets that need to be notified of a UI event; for asynchronous events this call is not needed. The sorting overhead has been removed to improve performance and also replaced internal structure to a TreeMap that maintains sorted order and reduces the cost of sorting.
Batching of DB	Saving items did not use JDBC batches when inserting data to the DB. Implemented

statements for Item Save	the use of JDBC batches to reduce network traffic and improve performance of inserting large number of the rows to the DB.

2.8.2 Completed change requests

Tracking Number	Description
16496	RunJob always returns code '0' which indicates success even when the job fails. Therefore the usage and the return code of RunJob has changed as follows: $JAVA_RT com.ibm.ccd.scheduler.common.RunJob: --job_name=aaaa --job_type=import|export [--company_code=bbbb --username=cccc --debug=on|off] The debug flag was added to option whether or not debugging information is to be printed. The return code is 0 if the job completes successfully, otherwise the return code is 1. If a given job is not found or if it cannot be started for any other reason then the return code is -1.
JR20960	Script compilation failing for script operations for certain type of scripts
15956	SQL error on Export Console screen
JR20876	getOldItem() does not return pre-workflow version of item being checked in
16526	Add a hierarchy map to an item, categories with no display name cannot be selected
16529	Perform a search and click the category link for an item, the category loads indefinitely

2.9 Fix Pack 5.0.1.3

2.9.1 Completed change requests

Tracking Number	Description
P13831	removeLocalesFromAttrGroup() fails
P14849	Faster population and validation and location-has-data
P14950	Rich Search - Can not search on unassigned items
P15125	Location Attributes - If error happens in another location hierarchy, clicking on the error will not bring user to the other hierarchy node
P15210	Missing operators e.g. beginswith/contains for lookup table attributes for global & item-location data
P15272	userDefinedLogEntryGetTarget() fails on a category
P15305	getFlatEntryNodes() returns duplicate locale root nodes
P15316	User Setting "In Multi Edit, show all category specific attributes..." gives widget error in workflow
P15351	getCatalogAttributes() does not return all the values set by setContainerProperties()
P15360	Single Edit screen – It is possible to add a new occurrence for an un-editable field
P15362	On sorting the date attributes of the Items, a server Error occurs
P15364	Lookup table item edit: exit button disappears when there is unsaved entry
P15414	**Provide gzip output stream support. (Refer to Section 2.4.2 for additional details)**
P15417	Defaulting the item-location hierarchy per workflow step

P15438	Search in background throws widget error in Workflow step
P15611	Second catalog get less items than the imported catalog
P15637	getLocaleNode() has the wrong prototype
P15706	Entry Preview Script made available to "**All Catalogs**" not available
P15720	Threw an exception on sorting (attribute type flag) of the results of Lookup table rich search
P15723	The user always has to press the refresh button to get the status of the checked out item/s
P15725	Sequence location attributes behave unexpectedly
P15786	Cannot see/edit location attributes in workflow or approval step
P15801	Primary hierarchy location attributes in workflow modify step
P15808	Rich Search on Relationship primary attribute type does not work for different conditions
P15833	Category restriction search (Rich Search) works only for the selected level and it's immediate child level
P15834	setInheriting() has the wrong documentation
P15835	new$SearchSelection is not supporting getSelectionItemCount()
P15848	runBackgroundSearch() does not create a new selection
P15854	Cannot access linked item within workflow
P15865	Secondary Hierarchy does not display any nodes in the left pane when expanded
P15885	Fail to retrieve implicit objects while checking out an item into collaboration area and while moving item to FIXIT
P15889	Imports run out of memory if too many are run in parallel
P15959	Cannot create gaps in inheritance of grouped location attributes
P15960	Inherited, grouped multi-occurrence location attribute values cannot be displayed
P16000	Icon appear to have data when data is missing
P16002	EntryNode.throwValidationError() throws error on all locations
P16004	Pre-processing script should not retrieve data value from the database
P16015/P15724	When checkout agreements to maintenance step it hangs
P16031	UI not being refreshed in workflow step
P16034	Workflow step finishes even though there is a validation error.
P16040	Workflow engine printing spurious error message when no events to process are found in DB
P16041	In a workflow step if you add a new item in the step using the UI and forget to put a primary key value, the item will not save as expected. Put in a value for the Primary keys after the failed save and save again a save error occurs.
P16042	Action menu item enable even if it has no actions defined
P16048	Bad DB2 SQL noticed in logs
P16052	Issue with EntryChangedData and location ops.
P16060	The "item selection" dropdown filter doesn't work for List View of Catalog module
P16063	EntryNode::hasInheritedValue() does not work
P16064	No logging of exception in module state server
P16065	Action menu item enable even if it has no actions defined
P16079	Displaying validation errors slow in location screen
P16081	Non-persistent attribute rules for location attributes need more context
P16097	**Added configuration options for running pre and postscripts for location attributes.**

	(Refer to section 2.4.3 for additional details.)
P16101	Get moniker method failed error pops up when logged in as new user other than admin and causes the Left Pane to not work properly
P16102	Item import to Collaboration area does not work
P16114	Select all 150 items in multi-edit causes some out of sync
P16121	Changes to Hierarchy nodes inside workflow are not reflected in the Left Pane
P16125	"Double quote" fails in string enumeration entry for a spec node
P16127	Re-catagorization within workflow does not work. (Replacing hierarchy mappings)
P16129	addLocalesToAttriGroup() fails
P16144	Vague Docs for getLocationsChangedToHaveData() and getLocationsChangedToHaveNoData()
P16145	Multi-edit for categories throws Java exception
P16150	Populate default value perf issue
P16151	getLocationChangedToHaveNoData() returns location even when the new entry has data
P16154	**Added new script operations for search templates. (Refer to Section 2.4.4 for additional details.)**
P16155	Do not perform save items if there are no items to save

2.9.2 P15414

Added support for a new element compress="yes" in the docstore_mount.xml file.

When set to "yes" anything written to the docstore will be stored gzipped (but no .gz extension is added) and the COMPRESSED attribute in the database will be set to indicate the file is compressed. Anything other than "yes", including no compress attribute, will cause the file to remain uncompressed to preserve existing behavior.

2.9.3 P16097

Added configuration options for running pre and post-scripts for location attributes

Running pre and post-script for each location can be a performance bottleneck. Now this is configurable in common.properties:

```
#run pre script for each available location when saving an item with
location attributes
run_pre_script_for_locations=true

#run post script for each available location when saving an item
with location attributes
run_post_script_for_locations=true
```

Specifying false for any of these parameters will disable running pre or post-script for location attributes when saving an item.

2.9.4 P16154

The following script operations for use with search templates were added to the script library:

- new SearchTemplate
- getSearchTemplateByName
- getSearchTemplateName
- deleteSearchTemplate

Refer to the Script Sandbox in WebSphere Product Center for documentation on the above script operations.

2.10 Fix Pack 5.0.1.2

Tracking Number	Description
P12363	Workflow events are not processed synchronously when nested scripts are called.
P12553	Auto-refresh of Item & Item-location screen when running macro and action scripts
P13662	Refresh and Undo buttons are not disabled in Rich Search screen
P14145	Location Attribute - Relationship value does not appear after specifying relationship and clicking Save
P14239	Checking out 150 items is not working
P14694	Standalone spec attributes do not appear when the spec is added to a hierarchy
P14949	Rich Search - If the same spec is used as a location for two hierarchies, the rich search screen attributes fields are not displayed properly
P14984	Only one collaboration area appears in the checkout tooltip, even when checked out to multiple areas
P15123	Checked out value rule attempts to update source on save and gives error if sent to interimCheckin
P15137	An error occurs when making a location available and saving after checkout
P15221	Saving invalid location data in workflow gives error asking if you want to navigate away from the page. This is the expected behavior.
P15239	Re-creating a company does not completely wipe out the old webs service data (intermittent).
P15271	Verify obsolete inheritance-related script operations
P15275	Population of offline search results container should be exposed via scripting
P15340	Errors in pre-processing script do not prevent checkin and do not add to the fixit step
P15341	Update documentation for hasInheritedValue()
P15342	arraylist concurrent modify error occurs when closing location data popup (doing a save)
P15345	Import into collaboration area works the first time only
P15348	New items added via import to workflow disappear on checkin
P15354	Item in source is not checked out if checked in from first collaboration area to which it has been checked out.
P15363	Location Attributes - java script error occurs when making a hierarchy node "available recursive"

P15374	Please remove item.setLocationAttrib(loc,path,value) OPS
P15375	Search is sometimes case sensitive. It should always ignore case.
P15380	entry.getEntrySaveResult() only return 'MODIFIED' entries
P15381	Entry::getDisplayValue() has wrong documentation or malfunctions
P15385	In the Item-location screen, using EntryNode::throwValidationError() does not display error messages
P15406	Script compilation fails for long return types
P15415	Cannot attach a secondary spec to a category
P15418	Entering value on locale attributes thrown error:entry can not be saved, but allows the item to move to the next step
P15419	Some popup windows are not sized optimally and need to be resized by the user
P15423	getLocationsAddedAvailability throws an error for first/one added location
P15424	Item-location inheritance does not fix issue with multi-occurrences
P15425	Nested grouping attribute loses data on save in workflow
P15426	Nested global groupings adding spurious occurrences on save in workflow
P15427	saveCtgItem() from wfl out() in() script throws class cast exception
P15429	When using the override button and saving the item, the attribute data disappears
P15431	Duplicate of P15426
P15432	Sub-attributes that are not editable in a grouping lose their data in a workflow
P15434	Entries with a minimum occurrence 1 group in location attributes get sent to the fixit step on checkin
P15543	Pressing "I" button throws an exception when there is nothing to inherit
P15544	collaboration_area, colArea, stepPath are returning NULL in Post Save script upon leaving the collaboration area
P15608	It is possible to bypass a reserve step in a hierarchy workflow using multi-edit
P15639	isInheriting()/hasNonInheritedValue/hasInheritedValue() are not working for Override with Null
P15683	Null overrides do not get saved/stored
P15711	It is not possible to create a tab view for hierarchy workflow

3. INSTALLATION, MIGRATION, UPGRADE, AND CONFIGURATION INFORMATION

This section provides general guidelines to apply a Fix Pack to WebSphere Product Center. Some information may differ depending on the methods used for previous installations. Contact your support representative for WebSphere Product Center with any installation issues.

3.1 Installation Notes

3.1.1 Configuration changes

3.1.1.1 Configuration changes for 5.0.1.8

Re-create company required

After the application of Fix Pack 5.0.1.8, it is required to create a new company to address the following issue:

17985 - Calling dropEntry() immediately after checkoutEntry() leaves ghost items in workflow. To remove the ghost entries that are already present in the existing database run the following script:

```
$TOP/src/db/schema/cmp/create_cmp.sh
```

The script will delete the existing company, if one exists, and create a new company.

Note: Create a full backup of the current WebSphere Product Center directories before deleting and creating a new company.

3.1.1.2 Configuration changes for 5.0.1.7

Schema changes

With the application of Fix Pack 5.0.1.7, the structure and the format of the two main database tables for storing items and categories was changed. The previous schema used will no longer work, therefore the schema must be updated.

To update the schema, run the script "create_schema.sh" to wipe out all the old schema data and create a new database schema. The script is located in the following directory:

```
$TOP/src/db/schema/create/create_schema.sh
```

Configuration changes

With the application of Fix Pack 5.0.1.7, the file **$DB2_HOME/cfg/db2cli.ini** must be updated to include the following line in the [COMMON] section:

```
PATCH2=50
```

Adding this line will free the LOB (Large Object) locators after reading each LOB. Otherwise, the following exception is thrown when the maximum (default 32000) number of LOB locators are used at the same time.

```
SQL0429N The maximum number of concurrent LOB locators has been
exceeded.
```

New common.properties parameter

The common.properties file now includes the configuration parameter called 'populate_location_default_values', which was added to improve item save performance. The value for this parameter should be set to 'false' if there are no default values, sequences or value rules for the location attributes. This reduces the time it takes to save an item with location attributes.

3.1.2 Support for document-literal style web services

This release includes support for Document-Literal style web services to the current web service functionality. The impact on migration from previous 5.0.1.x versions is the requirement of a minor database modification. Details for the database (DB2 and Oracle) modification are listed below:

Migration impact for DB2

For environments using DB2, run the following SQL:

```
create table twbs_wbs_webservice_temp
(
            wbs_id                        BIGINT not null,
            wbs_cmp_id                    BIGINT,
            wbs_name                      varchar(100) not null,
            wbs_desc                      varchar(2000),
            wbs_url                       varchar(2000),
            wbs_wsdl_url                  varchar(2000),
            wbs_wsdl_doc_path             varchar(100),
            wbs_protocol                  varchar(50),
            wbs_style                     varchar(50),
            wbs_impl_script_path          varchar(2000),
            wbs_store_incoming            char(1),
            wbs_store_outgoing            char(1),
            wbs_deployed                  char(1))
    IN USERS INDEX IN INDX;

INSERT INTO TWBS_WBS_WEBSERVICE_TEMP
    SELECT
            TWB.WBS_ID,
            TWB.WBS_CMP_ID,
            TWB.WBS_NAME,
            TWB.WBS_DESC,
            TWB.WBS_URL,
            TWB.WBS_WSDL_URL,
            TWB.WBS_WSDL_DOC_PATH,
            TWB.WBS_PROTOCOL,
            'RPC-Encoded',
            TWB.WBS_IMPL_SCRIPT_PATH,
            TWB.WBS_STORE_INCOMING,
            TWB.WBS_STORE_OUTGOING,
            TWB.WBS_DEPLOYED
    FROM
        TWBS_WBS_WEBSERVICE TWB;

COMMIT;

drop table twbs_wbs_webservice;

create table twbs_wbs_webservice
(
             wbs_id                        BIGINT not null,
             wbs_cmp_id                    BIGINT,
             wbs_name                      varchar(100) not null,
             wbs_desc                      varchar(2000),
             wbs_url                       varchar(2000),
             wbs_wsdl_url                  varchar(2000),
             wbs_wsdl_doc_path             varchar(100),
             wbs_protocol                  varchar(50),
             wbs_style                     varchar(50),
             wbs_impl_script_path          varchar(2000),
             wbs_store_incoming            char(1),
             wbs_store_outgoing            char(1),
             wbs_deployed                  char(1))
             IN USERS INDEX IN INDX;

ALTER TABLE TWBS_WBS_WEBSERVICE
        ADD CONSTRAINT WBS_0_PK PRIMARY KEY (WBS_ID);

ALTER TABLE TWBS_WBS_WEBSERVICE
        ADD CONSTRAINT WBS_1_UK UNIQUE (WBS_NAME);

INSERT INTO TWBS_WBS_WEBSERVICE
    SELECT
               TWB.WBS_ID,
               TWB.WBS_CMP_ID,
               TWB.WBS_NAME,
               TWB.WBS_DESC,
               TWB.WBS_URL,
               TWB.WBS_WSDL_URL,
               TWB.WBS_WSDL_DOC_PATH,
               TWB.WBS_PROTOCOL,
               'RPC-Encoded',
               TWB.WBS_IMPL_SCRIPT_PATH,
               TWB.WBS_STORE_INCOMING,
               TWB.WBS_STORE_OUTGOING,
               TWB.WBS_DEPLOYED
    FROM
        TWBS_WBS_WEBSERVICE_TEMP TWB;

COMMIT;


drop alias wbs;
create alias wbs for twbs_wbs_webservice;

--drop table twbs_wbs_webservice_temp;

commit;
```

Migration impact for Oracle

For environments using Oracle, run the following SQL:

```
create table twbs_wbs_webservice_temp
(
              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_2_pk           primary key (wbs_id),
             constraint wbs_3_uk           unique (wbs_name));

INSERT INTO TWBS_WBS_WEBSERVICE_TEMP
    SELECT
               TWB.WBS_ID,
               TWB.WBS_CMP_ID,
               TWB.WBS_NAME,
               TWB.WBS_DESC,
               TWB.WBS_URL,
               TWB.WBS_WSDL_URL,
               TWB.WBS_WSDL_DOC_PATH,
               TWB.WBS_PROTOCOL,
               'RPC-Encoded',
               TWB.WBS_IMPL_SCRIPT_PATH,
               TWB.WBS_STORE_INCOMING,
               TWB.WBS_STORE_OUTGOING,
               TWB.WBS_DEPLOYED
    FROM
        TWBS_WBS_WEBSERVICE TWB;

RENAME twbs_wbs_webservice to twbs_wbs_webservice_temp1;
RENAME twbs_wbs_webservice_temp to twbs_wbs_webservice;

--drop table twbs_wbs_webservice_temp1 cascade constraints;

alter table twbs_wbs_webservice drop constraint wbs_2_pk;
alter table twbs_wbs_webservice add constraint wbs_0_pk primary key
(wbs_id);

alter table twbs_wbs_webservice drop constraint wbs_3_uk;
alter table twbs_wbs_webservice add constraint wbs_1_uk unique
(wbs_name);

drop synonym wbs;
create synonym wbs for twbs_wbs_webservice;
```

3.1.3 Disable standard output logging for the WebSphere Application Server

Every time a patch is applied to WebSphere Application Server or WebSphere Product Center has been re-installed (Fix Packs and new installations) on WebSphere Application Server, it is required to disable standard output logging.

Note: This workaround address PMR 28535, which reports issues with logging into WebSphere Product Center due to a frozen WebSphere Application Server.

3.1.3.1 How to disable WebSphere Application Server standard output logging

Login to the WAS administration console, usually port 9090 or 9091.

From the left navigation pane select the following:

1) Change the JVM log settings

Servers/Application Servers

- The "Application Servers" page will appear in the main frame, select the following:

  <application name>

Logging and Tracing/JVM Logs

- The "JVM Logs" page should appear
- Under "General Properties" for both "System.out" and "System.err"
- Change the "File Name:" to "/dev/null" (make sure you type this correctly)
- Uncheck "Log File Rotation/File Size"
- Uncheck "Log File Rotation/Time"
- Set "Maximum Number of Historical Log Files" to 0
- Uncheck "Installed Application Output/Show application print statements"
- For "Installed Application Output" make sure you uncheck "Show application print statements"

2) Save the console changes

- Click apply
- Click on the "Save" URL from "Messages" window that appears
- Click on the "Save" button from the "Save to Master Configuration" menu that appears
- From the main menu go back to the "JVM Logs" console
- Click on the runtime tab
- Click "Apply"
- Click on the "Save" URL from "Messages" window that appears
- Click on the "Save" button from the "Save to Master Configuration" menu that appears

3) You can use the following UNIX command to verify that standard output logging has been disabled:

```
fuser $WAS_HOME/logs/$WAS_APPSERVERNAME/System*
```

- You should not see any numbers to the right of the ":" on any of the System files
- Please note that to use this verification method, you would need to first bounce the application after making the changes

3.2 Preparation

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

3.2.1 Stopping the whole application on the local machine

Complete the following steps to stop the WebSphere 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 following script:

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

2. Abort the entire application by running the following script:

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

   All services running on the local machine are 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.

3.2.2 Backup

- **Create a full backup of the current WebSphere 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 WebSphere Product Center version until performing thorough testing with the new installation.

3.2.3 Delete Tomcat working directory

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

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

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

3.3 Apply Fix Pack

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

- ❑ Unpack tar file
- ❑ Run WebSphere Application Server script
- ❑ Update configuration files
- ❑ Test installation

3.3.1 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 WebSphere Product Center files.

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

   Example:

   ```
   {HOME_OF_WPC}/tarballs
   ```

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

Example:

Using GNU tar, the following command extracts and unzips the tar file using an absolute path:

```
tar zxvf
/home/WPC/tarballs/wpc_5001_03_fixpak_from_5000_15_was5_db2.tgz
```

3.3.2 Run WebSphere Application Server script

After unpacking the tar file in the previous section, ensure that the default server (server1) is running and run the following WebSphere Application script:

```
$TOP/bin/websphere/install_war.sh
```

Note: Ensure that the default server (server1) is running, as it is required for the WebSphere Application Server script to work. If needed, start the WAS default server by issuing the following command as root:

```
${WAS_HOME}/bin/startServer.sh server1
```

3.3.3 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.

❑ common.properties

❑ admin_properties.xml

❑ init_ccd_vars.sh

❖ **common.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: $TOP/etc/default

❖ **admin_properties.xml**

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

File location: $TOP/etc/default

❖ **init_ccd_vars.sh**

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

File location: $TOP/setup

3.4 Test installation

3.4.1 Start WebSphere Product Center

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

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

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

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

3.4.2 Check status

Run the **$TOP/bin/go/rmi_status.sh** script that was provided by WebSphere 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 WEBSPHERE PRODUCT CENTER!

4. KNOWN LIMITATIONS, PROBLEMS AND WORKAROUNDS

NOTE: This section identifies issues and limitations that are known for this release and will be addressed in a future Fix Pack.

4.1 Required Location Attributes in a workflow are validated even when made unavailable

Case ID: 17133

Issue: In a workflow step if the user makes a location available then the required attributes for that location get validated when the item is saved. But if the user goes back and makes the location unavailable, the required attributes for that location still get validated in that step.

Workaround: The collaboration area Admin can go in that step and move the item to the Fixit step and then make the location unavailable.

4.2 Newly created Web services do not automatically deploy

Case ID: P16473

Issue: Create a new web service and restart WebSphere Product Center. An error appears when attempting to invoke the newly created web service.

Workaround: Allow write access to the Axis configuration file "server-config.wsdd" under the "public_html/WEB-INF" directory. Additionally, for environments using WebLogic, the WebSphere Product Center instance must be deployed in expanded directory format. If this is not done, the auto-redeploy functionality of Axis will not deploy the web services created by WebSphere Product Center on restart, thus causing an error.

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

Issue: Due to limitations caused by the XML parsing implementation (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 WEBSPHERE 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>
```

4.4 Cannot change the style when creating a new web service

Case ID: P16059

Issue: 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.

4.5 Entries disappear in interim checkin step

Case ID: 18128

Issue: Send an entry (item OR node) to an interim checkin step and it disappears.

4.6 Get missing primary key error when new node path is modified in workflow

Case ID: 18204

Issue: Get missing primary key error when new node path is modified in workflow. Add a new hierarchy node within a workflow for a hierarchy and save. Change the pathname and save. You get an error that the primary key is missing. The item can be re-accessed and saved with a new path. This happens even if the primary key and the path are different attributes.

4.7 Issues with logging into WebSphere Product Center due to a frozen WebSphere Application Server

Case ID: PMR 28535

Workaround: Every time a patch is applied to WebSphere Application Server or WebSphere Product Center has been re-installed (Fix Packs and new installations) on WebSphere Application Server, it is required to disable standard output logging.

Refer to Section 3.1.3.1 for instructions on disabling standard output logging on WebSphere Application Server.

5. DOCUMENTATION UPDATES

5.1 Java API Overview

The Java API and Java WebServices capabilities, implemented as part of this maintenance release, are two distinct layers of WPC capability. The base layer, the Java API, provides an Object Oriented mechanism to expose WPC artifacts, whilst the Java WebServices support provides the capability to develop and deploy java based WebServices written using the Java API.

Note: For this release only the java WebServices environment supports the use of the Java API.

There are two parts of the Java API:

- Build time component
- Runtime component

These allow a customer to develop Java based functionality on their client workstation utilizing their preferred development environment. The runtime component deliverable provides the realized Java API capabilities on the WPC server.

5.1.1 BuildTime component

The build time component JAR file contains only the exposed WPC JAVA API interfaces, factories and exceptions only. Within the initial release of the Java API it is not be possible to:

- Deploy user-developed code directly to the WPC J2EE server - Any customer-developed code, or web services, are deployed as User JARs.

- Utilize any integrated runtime environments to undertake testing - IDE's such as Rational Software Architect (RSA) have capability to install a local J2EE server which can be utilized to deploy, debug and test code developed within the IDE. No capabilities are provided to utilize the IDE specific local J2EE server as host for WPC J2EE capabilities. When utilizing an IDE such as RSA, users should have the ability to remotely debug their Java code once deployed to a test WPC server, should their deployment J2EE server support this capability.

5.1.2 RunTime component

The runtime component is integrated into the base WPC v5.0.1 environment. No additional user interaction is required.

5.1.3 Java WebServices

Note: For this release only the java WebServices environment supports the use of the Java API.

Support of the Java WebServices is provided via the internal WPC WebServices engine. Re-utilizing this capability provides a common WPC framework for deploying both Script and Java based webservices with WPC. Unlike Script based webservcies, there are several additional steps that must be undertaken to deploy a Java based webService. At a high level, these are:

- The java code making up the webservices must be bundled into a JAR file. Note that you can bundled the code for multiple java based webservcies into a single jar file

- The JAR file must then be deployed into the WPC environment, which is achieved by utilizing the deployment mechanism

- The WPC user utilizes the WPC UI to deploy the web service into the WPC environment

Unlike WPC Script based web services, it is possible to deploy a Java WebServices and find that the webservice fails when it is utilized. As all java-based webservices are deployed with JAR files, it is possible to accidentally remove the User JAR containing the WebService code. Therefore, care should be taken to ensure this does not occur, however should this be the case then the webservices fault would indicate this has occurred.

5.1.4 Java WebServices deployment and utilization

With proper configuration of WebSphere Product Center services and web application server, Java Webservices can securely pass information through to an external link,

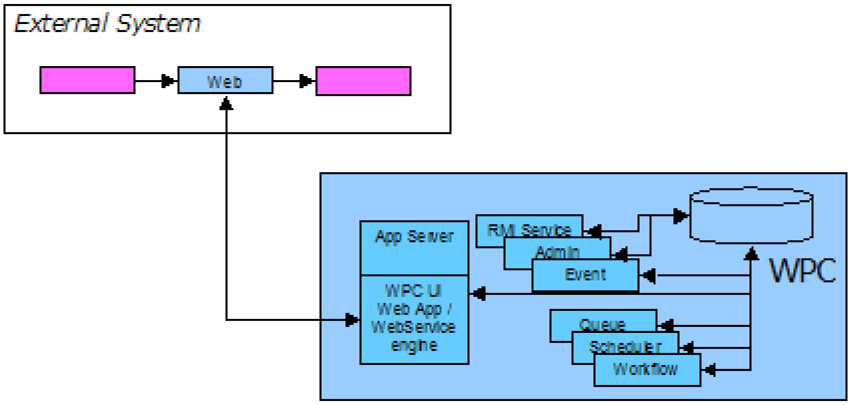


Figure 1: Java WebService

Java WebServices can be deployed into a simple single WPC instance environment, see Figure 2: Single WPC Server , and larger complex scaled environments, see Figure 3: Scaled WPC Server environment.

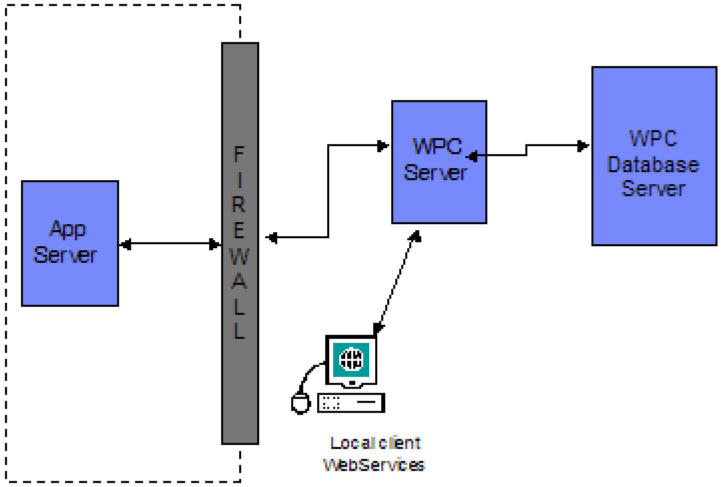


Figure 2: Single WPC Server environment

When utilizing a larger scaled WPC environment it is the responsibility of the WPC administrator to ensure that the User Jar file containing the Java WebServices is deployed on all systems where the web services are expected to execute. This is especially important when a shared, or replicated, database is utilized, as the availability of a web service is control by an entry into the WPC database. Utilizing a shared database will make web services available within all WPC WebServices environments that use the database.

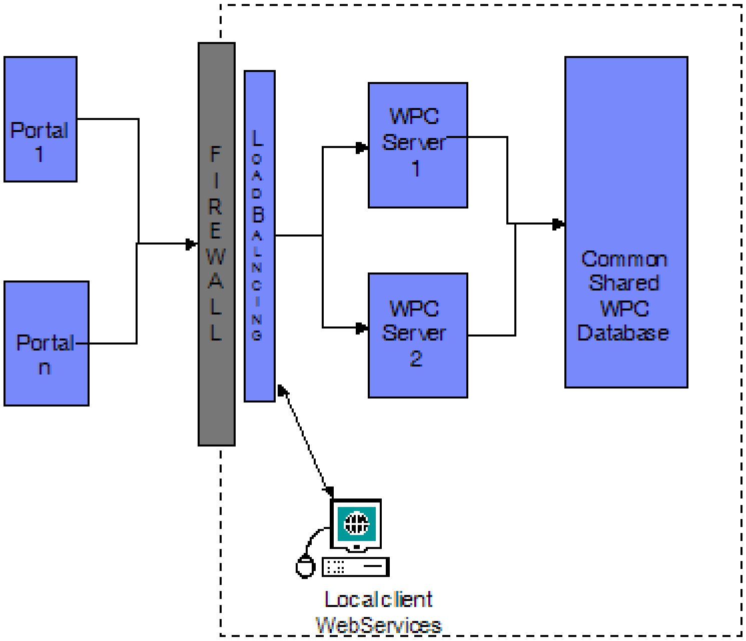


Figure 3: Scaled WPC Server environment

When using Java based webservices, there are several restrictions to note:

- All WebServices are stateless. It is the responsibility of the WPC customer to ensure any state information required across multiple WebServices call is persisted. There are several mechanisms to achieve this including User Defined Logs and persisting out to a database by the use of the JDBC.
- No standalone WebServices testing framework is provided
- No WebServices serialisers or deserialisers are provided for WPC artifacts in this release
- There is no support for attachments within java-based webservices

5.2 Using the Java API

5.2.1 Limitations in the WPC Java API

The Java API provides a limited access into WPC product centre. Not all objects are supported and in the majority of cases only read access are provide to the objects exposed. Full Create, Read, Update and Delete capabilities exist for WPC Items and User Defined Logs. Please refer to the Java API java doc for a full list of supported objects and their capabilities.

5.2.2 Using the Java API

The methodology to undertake any functionality within the Java API is similar regardless of the type of task the user is attempting to code. Here we demonstrate the use of the Java API to create a Java API search and process the results.

```
//Obtain a PIM Context and a Search Manager
try
{
   ctx = PIMContextFactory.getContext(USERNAME, PASSWORD, 
COMPANY_NAME);
   searchManager = ctx.getSearchManager();
}
catch (Exception e1)
{
   System.out.println("Exception " + e1 + " = " + e1.getCause());
}


// Obtain a Search query object
String queryString = "select item from catalog('SearchAPICatalog') 
where {BasicAPISpec/PK} = \"ITEM1\"";

SearchQuery query = searchManager.getSearchQuery(queryString);

// Execute the query
SearchResultSet resultSet = query.execute();

// Process the Search result Set
String PK = null;
while (resultSet.next())
{
   PK = resultSet.getString(1);
   System.out.println("Primary Key: " + PK + "found");
}
```

As with all capability within the Java API, the user must first retrieve the Java API context object. As shown in xxxx, when creating a context the developer must specific a WPC user, their password and associated company. After retrieving the context, the developer must retrieve one of the several supported Java API Managers to access the required capabilities. In the example bellow we can see that the developer is retrieving the Java API search manager to undertake a search, please refer to the Java API Java doc for information on the available Java API managers.

Having retrieved an instance of a Search Manager, the developer can access the search specific methods, in this case the getSearchQuery() method to return a search object, which can be used to undertake the search. It should be noted that in WPC V5.0.1 only WPC Item searches are supported. As can be seen in the example, we specify the query, for WPC V5.0.1 there is a strict syntax for the query string and it must follow the rules specified bellow:

1) Start with "select item from catalog('"

2) Have a catalog name that is 1 or more characters in length

3) This must be followed by "') where '"

4) This must be followed by a where clause expression that matches the existing expression for the runSearch script operation.

Having retrieved a query the user can then use the execute() method to run the query and process the returned Search Result Set as shown. The Java API search capability has been designed to follow the same approach as JDBC queries and result set processing.

All previous stated all Java API code usages follows the same model of retrieving a WPC context, and then using the various managers to gain access to the WPC artifacts. Please refer to the Java doc for a full list of managers and objects currently provided with the WPC Java API.

5.2.3 Javadoc location

It access the java documentation:

1) Untar the install tgz file and in the a folder "javaapi", there contains two files: ccd_javaapi.jar and ccd_javaapidoc.zip

2) unzip the ccd_javaapidoc.zip file, to view the HTML files

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.