



IBM Global Data Synchronization for WebSphere Product Center
Version 1.2

Customization Guide

Note: Before using this information and the product it supports, read the information in “Notices” on page 77.

29 July, 2005

This edition of this document applies to IBM Global Data Synchronization for Websphere Product Center, version 1.2, and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporations 2005. All rights reserved. Licensed Materials-Property of IBM

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



ABOUT THIS DOCUMENT.....	5
AUDIENCE.....	5
RELATED DOCUMENTATION.....	5
TYPOGRAPHIC CONVENTIONS.....	5
CH 1 MASS TRANSACTIONS	6
<i>Search filter:</i>	6
<i>Creating a Mass transaction report</i>	6
<i>Mass Global Enrichment</i>	7
<i>Mass Local Enrichment</i>	7
<i>Mass Compliance check</i>	8
<i>Mass Submission for item approval</i>	9
<i>Mass Item Add</i>	9
<i>Mass Publish Items</i>	10
<i>Mass synchronization of Items</i>	10
<i>Mass Add Item Link</i>	11
CH 2 MASS UPDATE	12
CH 3 CONFIGURING CUSTOM WORKFLOWS	16
CH 4 IMPORT/EXPORT – SUPPLY SIDE.....	18
FILE FORMATS FOR IMPORT /EXPORT	18
<i>Implementation Approach for CSV</i>	18
<i>Implementation Approach for XML</i>	19
CATALOG TO CATALOG IMPORT	19
SCENARIOS.....	19
<i>For Import</i>	19
HANDLING VALIDATIONS.....	26
<i>Export</i>	26
SCRIPT TABLE.....	29
<i>Script operations related to Trade Item</i>	30
<i>Script operations related to Trade Item Link</i>	34
<i>Script operations related to Trading Partner</i>	35
<i>Script operations related to Notifications</i>	36
<i>Script operations related to Transactions</i>	36
USING SCRIPT OPERATIONS.....	37
<i>Trade Items</i>	37
<i>Trading Partners</i>	41
<i>Trade Item Links</i>	43
<i>Spec Maps</i>	45
CH 5 IMPORT/EXPORT – DEMAND SIDE.....	46
FILE FORMATS FOR IMPORT /EXPORT	46
<i>Implementation Approach for CSV</i>	46
<i>Implementation Approach for XML</i>	47
SCENARIOS.....	47
<i>For Import</i>	47
HANDLING VALIDATIONS.....	53
<i>Export</i>	54
SCRIPT TABLE.....	57
<i>Script operations related to Trade Item</i>	57
<i>Script operations related to Trade Item Links</i>	60
<i>Script operations related to Trading partner</i>	61



<i>Script operations related to Notifications</i>	62
<i>Script operations related to Transactions</i>	62
<i>Script operations related to Messaging</i>	63
<i>Script operations related to Spec Map</i>	70
USING SCRIPT OPERATIONS.....	70
<i>Trade Items</i>	70
<i>Trading Partners</i>	75
<i>Trade Item Links</i>	76
<i>Spec Maps</i>	78
NOTICES	80



About This Document

Audience

The objective of this guide is to facilitate the customization of the IBM Global Data Synchronization for WebSphere Product Center solution for supply and demand components.

The intended audiences for this guide are the system administrators and the IT analysts.

Related Documentation

The following related Global Data Synchronization Demand documentation is also available:

- User Guide
- Installation and Configuration Guide
- Release Notes

Typographic Conventions

This document uses the following conventions:

- `Courier New font` - Indicates commands, directory names, user names, path names, and file names.
- **Note** - Indicates important or clarifying information including exceptions, interdependencies, and special situations.
- **Tip** - Indicates a helpful action that will simplify or streamline the current procedure.
- **bold font** - Indicates buttons, tabs, menus, and keyboard key names. For example, the **Save** button.
- *Italics font* - Indicates screen names. For example, the *Login* screen.

Ch 1 Mass Transactions

The mass transactions are scheduled jobs in Websphere Product Center. All the jobs are defined as reports in WebSphere Product Center. These transactions are specific to supply side.

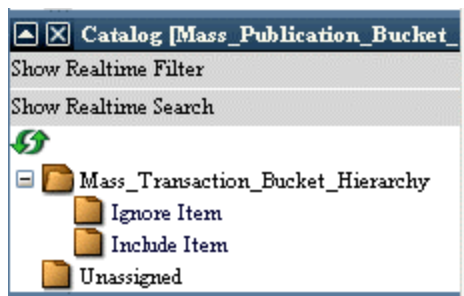
While executing these jobs, the user can specify a search criterion. Each report has a set of search parameters in the inputs parameter spec for the report. Using this you will be able to search for trade items that would be submitted for the various mass transactions.

Search filter:

The user can add or ignore the trade items in the search results.

To add or ignore the trade items in the search results:

Create an item in "Mass_Publication_Bucket_Catalog". The catalog has a Mass_Transaction_bucket_hierarchy which has two categories, Ignore Item and Include Item.



Depending on the category that you choose, the script will construct trade items using the values (GTIN, TM & IP) and include or remove it from the search results.

Creating a Mass transaction report

To create the report:

Go to Product Manager> Reports>New Report.

From the report type dropw down, select the type of the report.

The different types of mass transactions that are preseeded in the application are:

- Mass Global Enrichment
- Mass Local Enrichment
- Mass Compliance Check
- Mass Submission for Item Approval
- MassAdd Item

- Mass Publish Items
- Mass Synchronization of Items
- Mass Add Item Link

In the report name field, enter a name of the report.

From the distribution type drop down select the type of the distribution.

This will complete creating a new report.

Mass Global Enrichment

This job will submit all the selected trade items into the global enrichment workflow. These selected trade items should be in “draft” state.

To run a Mass Global enrichment check:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose the report of type Mass Global enrichment.



In the list of fields displayed, enter the search criteria for the items that you want to submit for global enrichment. If no value is specified all items that are in draft state are picked up.


Note:

The user eligible for Global-Local Enrichment will get one notification for each trade item submitted for Global-Local Enrichment.

After completion of Global-Local Enrichment for each Trade Item, a particular user, for example Admin, will be getting the Enrichment Completion notification. Hence the security for this user has to be configured. For more details on security configuration refer to *chapter 3 Security Configurations* in the *Installation and Configuration guide*.

One can see the status of the Trade Item which has been submitted for Global-Local Enrichment through GDS Reports. The user should be able to view using “Explore Transactions”.

Save the report by clicking on the **Save**  icon and go back to the report’s console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Mass Local Enrichment

This job will submit all the selected trade items into the local enrichment workflow. The status of the selected trade items should draft with variant.

To run Mass Local enrichment:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose a report of type Mass Local enrichment.

In the list of fields displayed, enter the search values for the items that you want to submit for local enrichment. If no value is specified all items that are in draft state are picked up.


Note: The user eligible for Global-Local Enrichment will get one notification for each trade item submitted for Global-Local Enrichment.

After completion of Global-Local Enrichment for each Trade Item, a particular user, for example Admin, will be getting the Enrichment Completion notification. Hence the security for this user has to be configured. For more details on security configuration refer to *chapter 3 Security Configurations* in the *Installation and Configuration guide*

One can see the status of the Trade Item which has been submitted for Global-Local Enrichment through GDS Reports. The user should be able to view using “Explore Transactions”.

All the variants of a trade item matching the criteria will be picked up by the job.

Save the report by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Mass Compliance check

This job will perform a compliance check for the selected trade items. The selected trade items should be in “complete”, “edited” or “modified locally” state. This job will generate a report for both the failed and passed trade items in the docstore.

To run a Mass compliance check:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose a report of type Mass compliance check.

In the list of fields displayed, enter the search values for the items that you want to submit for mass compliance check. If no value is specified items which are in “Complete” state are picked up.

Save the report by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report click on the scheduler  icon.

Note: The log generated will give the list of Items that have failed compliance check with description of particular attributes that have failed validations. The log will also give the list of items passed the compliance check

Mass Submission for item approval

This job will submit all the selected trade items into item approval workflow. These selected trade items should be in “compliant” state.

To run a Mass Item approval:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose a report of type Mass submission for item approval.

In the list of fields displayed, enter the search values for the items that you want to submit for approval. If no value is specified then all trade items that are compliant are picked up and submitted for approval.

Note: The user eligible for approval will get one notification for each Trade Item submitted for approval.

After approval, a particular user, for example Admin, will be getting the approve/Reject notification. The security of the user has to be set properly.

One can see the status of the Trade Item which has been submitted for Approval through the Reports menu in the IBM Global Data Synchronization for WebSphere Product Center, user interface.

Save the report by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Mass Item Add

This job will generate an XML with the list of items to be submitted to the data pool. The XML can be viewed in the docstore archive. These selected trade items should be in “approved” state.

To run a Mass Item Add:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose a report of type Mass Item Add.

In the list of fields displayed, enter the search values for the items that you want to publish. In no value is specified, the report picks all the items which have successfully completed compliance check and are in “approved” state.

Save the report by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Mass Publish Items


This job will generate an XML for initial load or new item, with the list of items to be published to the trading partner. The XML can be viewed in the docstore archive. The selected trade items should be in registered, registered-GS1, registered-GS1 non compliant state.

To run a Mass Publish Items:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose a report of type Mass Publish Items.

In the list of fields displayed, enter the search values for the items that you want to publish. If no value is specified then items with global status as “registered/registere-GS1” and local status as “registered/registered GS1” are picked up.

Note: It is mandatory to enter values in the fields that are marked  .

Note: In the trading partner field you can enter multiple values separated by a comma “,”.

Note: The publication date is the date on which the publication will be sent by the data pool to the trading partners.

Note: Mass Publication can be either an “initial load item publication” or “new item publication”

Note: If the field cascade item is set to true, then the item will be published with the entire hierarchy.

Save the report by clicking on the **Save**  icon and go back to the report’s console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Mass synchronization of Items

This job will generate an XML with the list of items to synchronise. The XML can be viewed in the docstore archive. These selected trade items should be in “modified locally”state.

To run mass synchronization of items:

Go to Product Manager> Reports> Reports Console.

From the list of transactions, choose a report of type Mass Synchronization of Items.

In the list of fields displayed, enter the search values for the items that you want to add to synchronize. If no value is specified then all trade items that have been “modified locally” are picked up for synchronization.

Save the report by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Mass Add Item Link

This job will generate an XML with the list of item links to be submitted into the data pool. The XML can be viewed in the docstore archive. The selected trade items link should be in “approved” state.


To run a Mass Add Item Link report:

Go to Product Manager> Reports> Reports Console.

From the list of reports, choose a report of type Mass Add Item Link.

In the list of fields displayed, enter the search values for the items that you want to add item links to.

Save the search parameters by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Note: The log generated by all the above reports can be viewed in the document store. To access the report go to Collaboration Manager> Document Store> reports> “Name of the report”. The report name is represented by the time stamp of the time that the report was executed.

Ch 2 Mass Update

Through this functionality, the user will be able to update a set of GTINs at one point of time on supply side.

To execute a mass update job:

Go to Product Manager> Reports> Reports Console


To create a new report:

Click the **New**  icon.

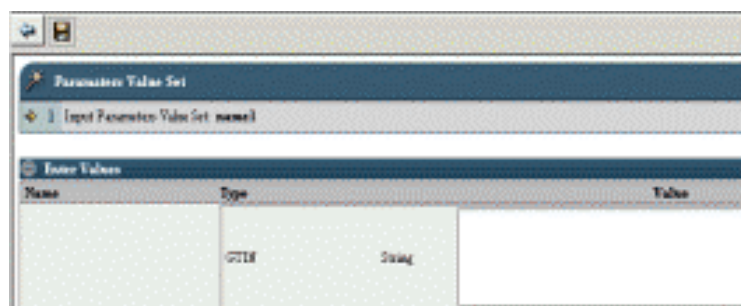
Select the report type as Mass update.

In the report name field, type in the report name and click on **Next**.

Select the distribution.

Click on  to go back to the reports console.

From the list of reports, choose the report that you want to run. The page to enter values for the attributes is displayed.



In the list of fields displayed, under Mass Update Search criteria enter the search values for the trade items that you want to add to the data pool. You have to mandatorily enter at least one search criterion.

Under the Mass Update Global attributes and Mass Update Local Attributes, enter the values for the attributes for which you want to perform a mass update.

Under the Mass Update Spec custom attributes, you can update the value by specifying the complete path of the attributes.

For example: Global_Attributes_Spec/Brand/BrandName

Note: The language specific attributes, interoperability attributes and partner specific attributes can be updated by the custom attributes provided

Note: You cannot perform a massupdate for the following attributes: GTIN, TM, IP, Product type, Internal Category, Data Pool Category, GPC classification, EAN UCC code, EAN UCC type.

Save the report by clicking on the **Save**  icon and go back to the report's console by clicking on the  icon.

To schedule the report, click on the scheduler  icon.

Note: If an item is stuck in a workflow then a mass update will not be performed on that item.

After completing a mass update a trade item goes through the following status changes:

Local Status	Status Transition after Mass Update
Approved	Compliant
Compliant	Compliant
Edited	Compliant
Modified Compliant	Compliant
Modified Locally	Modified Locally
Registered	Modified Locally
Registered GS1	Modified Locally
Registered GS1 – Non Compliant	Modified Locally

Script for mass jobs

This following script operation is a single entry point for all the Mass Jobs. The type of mass job is identified by the ID.

Operation name and signature:

```
processMassTransaction_SS(String massJobID,HashMap hmSearchParamsToValues)
```

Description:

A valid Mass Job ID can be anyone of the following values

Mass Job Type	ID
Global_Enrichment	1



Local_Enrichment	2
Compliance_Check	3
Item_Approval	4
Add_Item	5
Add_Item_Link	6
Publish_Item	7
Synchronize_Changes	8

The user can also pass the search parameters to via hash map. The valid keys are as follows and they are case sensitive.

1. GTIN
2. GTIN_NAME
3. BRAND_NAME
4. PRODUCT_NAME
5. PRODUCT_DESCRIPTION
6. TARGET_MARKET
7. IP_GLN
8. IP_NAME
9. MANUFACTURER_GLN
10. MANUFACTURER_NAME
11. PRODUCT_TYPE
12. INTERNAL_CODE
13. INTERNAL_CODE_DESC
14. UDEX_CODE
15. UDEX_CATEGORY_DESC

The following is an example of how the mass job can be invoked through the WebSphere Product Center script sandbox.

```
//initialize the mass job type
var massJobID = 1; //Global Enrichment

//populate the search parameter values
hmSearchParamsToValues["GTIN"]="0001230123012" ;
hmSearchParamsToValues["TARGET_MARKET"]="US" ;
hmSearchParamsToValues["IP_GLN"]="000123929234" ;

//invoke the mass transaction
```



```
processMassTransaction_SS  
(massJobID,hmSearchParamsToValues)
```

Mass jobs performance enhancements:

Ensure the following setting in order to ensure better performance of the mass import export operations:

At the starting of the mass import/export script:

```
setMassImport_SS(toBoolean("true"));
```

At the end of the mass import/export script:

```
setMassImport_SS(toBoolean("false"));
```


Ch 3 Configuring custom workflows

For demand side publications the following are the default workflows that are invoked in case of normal flow in the IBM global Data synchronization for Websphere Product Center application

Normal Workflows:

- NEW ITEM - ProcessNewItemWF
- INITIAL LOAD - ProcessInitialLoadWF
- ITEM CHANGE - ProcessItemChangeWF
- ITEM DELIST – ProcessItemDelistWF *[only in Uccnet]*
- ITEM WITHDRAW – ProcessItemWithdraw WF *[only in Uccnet]*
- ITEM CORRECT - ProcessItemCorrectWF *[only for WWRE]*

In order to invoke a custom process flow for Catalogue Item Notifications [CIN] (for example: where the external legacy system does item management), a configuration file has been provided to specify the custom workflows for different CIN types.

Following is the location of this configuration file:

\$STOP/etc/processflow/<datapool name>/publicationprocess.properties

Example: \$STOP/etc/processflow/uccnet/publicationprocess.properties

The following constants can be used to configure the custom workflow names

- For New Item: CIN_NEW_ITEM_PROCESS
- For Initial Load: CIN_INITIAL_LOAD_PROCESS
- For Item Change: CIN_ITEM_CHANGE_PROCESS
- For Item Withdraw: CIN_ITEM_WITHDRAW_PROCESS
- For Item De-list: CIN_ITEM_DELIST_PROCESS
- For Item Correct: CIN_ITEM_CORRECT_PROCESS

Sample configuration file:

```
#NEW ITEM
CIN_NEW_ITEM_PROCESS=ProcessNewItemWFCustom

#INITIAL LOAD
CIN_INITIAL_LOAD_PROCESS=ProcessInitialLoadWFCustom

#ITEM CHANGE
CIN_ITEM_CHANGE_PROCESS=ProcessItemChangeWFCustom
```



The above sample defines custom workflows for New Item, Initial Load and Item Change publications. Any publication of the above types will be checked-out in the custom workflows instead of the normal GDS workflows.

With this configuration it is very easy to switch between the normal GDS CIN workflows and any custom workflow. In case the *publicationprocess.properties* file cannot be loaded, the default GDS workflows (as mentioned under the *Normal Workflows* section) will be invoked for the publications.

Ch 4 Import/Export – Supply Side

The Import/Export operation allows the user to export or import information from or to IBM Global Data Synchronization for WebSphere Product Center. The user can import or export information about trade items, trade item links, and trading partners. This chapter provides a few guidelines for the PSS personnel while implementing IBM Global Data Synchronization for WebSphere Product Center for the customer. The various scenarios for import/export are explained.

Note: IBM Global Data Synchronization for WebSphere Product Center does not have separate import/export modules. For release 1.2, WebSphere Product Center 5.1.2.0-IF2 is used for import/export.

The import or export depends a lot on the data available at the customer location. Some may not have GTIN as the primary key for trade items, while for some the mandatory fields may be missing. In such cases, the PSS team should setup the import or export process correctly.

If the GTIN is not present in legacy systems, a GTIN generator needs to be plugged in while importing the data from legacy systems. The generated GTIN need not be stored in the legacy systems and can be dynamically generated. Ensure that the generated GTIN is unique. It should be the same every time the data is imported or exported and not be a randomly generated number.

If the source data does not contain mandatory fields (required by IBM Global Data Synchronization for WebSphere Product Center), a dummy value needs to be generated. The trade items can be modified after the import.

The imported trade item status should be compliant with the status defined in IBM Global data synchronization for WebSphere Product center, [refer trade item status table](#) below on page 18. This can be done while performing import or export by modifying the import/export script.

Once the trade items are imported, the user can ensure the datapool compliance, by running the compliance script provided.

File Formats for Import/Export

The data can be imported or exported in different file formats. As a sample, imports or exports are provided in CSV (Comma Separated Values) and XML formats. This can be extended to suite the customer's requirements.

Implementation Approach for CSV

The basic trade item is represented by two specs:

- Global_Attributes_Spec
- Global_Local_Attributes_Spec.



When setting up an export operation, a destination spec needs to be defined that maps to these two specs. The destination spec, maps the attribute names of the trade items in the legacy system to the attribute names of trade items in IBM Global Data Synchronization for WebSphere Product Center.

Similarly, when setting up an import, a file spec needs to be defined that maps to these two specs. The file spec maps the attribute names of the trade items in IBM Global Data Synchronization for WebSphere Product Center to the attribute names of trade items in the legacy systems.

Implementation Approach for XML

Using the IBM Global Data Synchronization for WebSphere Product Center Business Object script operations, the xml/csv file can be imported into IBM Global Data Synchronization for WebSphere Product Center. The script operations are explained in the following sections.

Catalog to Catalog Import

For information about catalog to catalog import, refer the *WebSphere Product Center Version 5.2.0-IF2 User Guide*.

Scenarios

This section lists down the various customer scenarios for import and export, using the various script operations provided by both IBM Global Data synchronization for WebSphere Product Center and WebSphere Product Center.

Note: For information o WebSphere Product Center script operations, refer the *WebSphere Product Center 5.2.0-IF2 User's Guide*.

Note: For more information on individual script operations in IBM Global Data synchronization for WebSphere Product Center, refer to the section [using script operations](#)

For Import

This section explains the various scenarios and gives an overview of the import scripts.

Note: When an item is imported through scripts, unlike the IBM Global Data synchronization for WebSphere Product Center user interface, it does not go through an approval process. All approval processes are implicit and the user has to set the appropriate final status for the trade item. For the list of status changes refer to the [trade item status list](#) table below.

- [Import Global Trade Items](#)
- [Import Global and Local Trade Items](#)
- [Import Trade Item in a Particular Status](#)

- [Update an Already Existing Trade Item through Import](#)
- [Import Trade Item Links in a Particular Status](#)
- [Updating Trade Item Links through Import](#)
- [Import Trading Partners](#)
- [Updating a trading partners through import](#)

Import Global Trade Items

For this scenario a specmap is created in the system which maps the xml attributes to the Global_Attributes_Spec of the Global_Catalog. Then use the script function specMap.getMappings () which returns a hashmap of the destination paths to the source paths. Using this, the xml nodes are parsed and the corresponding attribute path values are set.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the Global_Attributes_Spec of the Global_Catalog. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec map.
- Create a trade item business object.
- Set the obtained attribute values on the trade item business object.
- Set the appropriate status for the global item in the trade item business object.
- Save the trade item business object.
- Create a transaction log for the imported trade item.
- Notify the user by creating an appropriate notification.(optional)

Global Item status	Local Item status
Draft	Not applicable
Global enrichment completed	

Import Global and Local Trade Items

This script will import the trade item with a variant, consisting of global item and global local item attributes into the Global_catalog and Global_local_catalog respectively.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the Global_Attributes_Spec of the Global_Catalog and Global_Local_Attributes_Spec of the Global_Local_Catalog. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Create a trade item business object.
- Set the obtained attribute values on the trade item business object.
- Set the appropriate status for the global item and the local item in the trade item business object.
- Depending on the current status of the trade item, ensure that the trade item passes the compliance check.
- Save the trade item business object.
- Create a transaction log for the imported trade item.
- Notify the user by creating an appropriate notification.(optional)

Update an Already Existing Trade Item through Import

This script will import the trade item with a variant, consisting of global item and global local item into the Global_catalog and Global_local_catalog respectively.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the Global_Attributes_Spec of the Global_Catalog and Global_Local_Attributes_Spec of the Global_Local_Catalog. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Get the handle of the trade item business object.
- Set the obtained attribute values on the trade item business object.
- Set the appropriate status for the global item and the local item in the trade item business object.
- Depending on the current status of the trade item, ensure that the trade item passes the compliance check.
- Save the trade item business object.
- Create a transaction log for the updated trade item.
- Notify the user by creating an appropriate notification.(optional)



Trade Item Status Table

The following table shows the rules that govern status of trade items.

For Global Item:

From status: Status of the trade item prior to import

To Status: Status of the item after import

Compliance check required: Specifies whether the trade item has to go through a compliance check

Notification: The notification column specifies whether the user has to be sent a notification or not. The types of notification are defined in the NotificationSecurityMappingTable lookup table.

Transaction: This column specifies which type of transaction needs to be logged once the import is done. Transaction types are defined in Transaction_type lookup table.

From Status	To Status	Compliance Check Required	Notification	Transaction
	Draft			CREATE_GTIN
Draft	Draft	NO		
Draft With Variant	Draft With Variant	NO		
Draft with Attribute	Global Attributes Modified Locally	NO	MODIFY_ITEM_APPROVED	MODIFY_GLOBAL_ITEM_APPROVED
Global Attributes Modified Locally	Global Attributes Modified Locally	YES		
Global Enrichment Completed	Global Enrichment Completed	NO		
Pending Variant Creation	Pending Variant Creation	NO		
Modified Compliant	Modified Compliant	YES		
Modified Compliant Approval Pending	Modified Compliant Approval Pending	YES		



From Status	To Status	Compliance Check Required	Notification	Transaction
Registered	Modified Locally	YES	MODIFY_ITEM_APPROVED	MODIFY_ITEM_APPROVED

For Local Item

From Status	To Status	Compliance Check Required	Notification	Transaction
Draft	Draft	NO		
Modified Compliant	Modified Compliant	YES		
Modified Compliant Approval Pending	Modified Compliant Approval Pending	YES		
Modified Locally	Modified Compliant Approval Pending	YES	MODIFY_ITEM_APPROVAL_PENDING	MODIFY_ITEM_APPROVAL_PENDING
Complete	Edited	YES		
Compliant	Edited	YES		
Compliant	Compliant	YES		
Approved	Edited	NO		
Rejected	Compliant	YES		
Registered	Modified Locally	YES	MODIFY_ITEM_APPROVED	MODIFY_ITEM_APPROVED

Import Trade Item Links

This script will import the trade item links into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the GTIN_Hierarchy_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Create a trade item link business object.



- The link should have valid combinations of the product types for the parent and the child GTIN.
- Set the obtained attribute values on the trade item link business object.
- Set the appropriate status for the trade item link business object.
- Save the trade item link business object.
- Create a transaction log for the imported trade item link.
- Notify the user by creating an appropriate notification.(optional)

Update Trade Item Links through Import

This script will update the trade item links into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the GTIN_Hierarchy_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Obtain a trade item link business object.
- The link should have valid combinations of the product types for the parent and the child GTIN.
- Set the obtained attribute values on the trade item link business object.
- Set the appropriate status for the trade item link business object.
- Save the trade item link business object.
- Create a transaction log for the imported trade item link.
- Notify the user by creating an appropriate notification.(optional)

The following table shows the rules that govern the status of trade item links.

From_Status	To_Status	Notification	Transaction
Draft	Draft		
Approved	Edited	MODIFIED_ITEM_LINK_APPROVAL_PENDING	MODIFIED_ITEM_LINK_APPROVAL_PENDING
Rejected	Edited	MODIFIED_ITEM_LINK_APPROVAL_PENDING	MODIFIED_ITEM_LINK_APPROVAL_PENDING
Registered	Modified Locally	MODIFIED_ITEM_LINK_APPROVED	MODIFIED_ITEM_LINK_APPROVED
Modified Locally	Modified Locally		



Import Trading Partners

This script will import the trading partners into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes GLN_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Create a trading partner business object.
- Set the attribute “item type” as “Trading Partner” on the trading partner business object.
- Set the obtained attribute values on the trading partner business object.
- Set the appropriate status for the trading partner business object.
- Save the trading partner business object.
- Create a transaction log for the imported trading partner.
- Notify the user by creating an appropriate notification.(optional)

Update Trading Partners through import

This script will update the trading partners into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes GLN_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Obtain a trading partner business object.
- Set the obtained attribute values on the trading partner business object.
- Set the appropriate status for the trading partner business object.
- Save the trading partner business object.
- Create a transaction log for the imported trading partner.
- Notify the user by creating an appropriate notification.(optional)

The following table shows the rules that govern auto status of trading partners.

Existing Status	After Import	Notification	Transaction
Draft	Draft		0
Approved	Edited		0
Rejected	Edited		0



Existing Status	After Import	Notification	Transaction
Edited	Edited		0

Handling Validations

Three validation scripts have been exposed to check if the trade item is compliant with an exchange. They are.

1. `HashMap TradeItem::validateTradeItem()`

This method validates all the attributes for a `TradeItem` object and returns a `HashMap` of error messages only for those attributes which have failed validation.

2. `String TradeItem::validateTradeItemAttribute(StringsAttrPath)`

This method validates the attribute for a `TradeItem` object and returns an error message if the attribute has errors else it returns null.

3. `HashMap TradeItem::validateTradeItemAttributeGroup(StringsAttributeGroup)`

This method validates an attribute group for a `TradeItem` object and returns a `HashMap` of error messages only for those attributes which have failed validation.

Based on the business requirement these functions can be called in the import script before saving the trade item.

Export

Note: All the Export scripts refer to a library script called `GDSLlibrary.GenerateXML`. This script basically generates the xml definition for trade item, trading partners, and trade item links. This script is as shown.

The various scenarios for export are:

- [Export trade items that have been created or updated in a date range](#)
- [Export Trade Items which are in a Particular Status](#)
- [Export Trade Items Based on Local Status](#)
- [Export Trade Items Based on Particular Category](#)
- [Export Trade Items Based on a search criteria](#)
- [Export All Trade Item Links](#)
- [Export Trade Item Links Based in status](#)
- [Export trading Partner](#)

- [Export Trading Partner – Search Based](#)

Export Trade Items that have been created or updated in a date range

This script will export the trade items data into XML format that has been created or updated between a certain date range.

Overview of the script:

- Get the list of the trade items depending on a particular transaction type using the `getModifiedTradeItems()` script operation
- Generate an XML output for each trade item.

Export Trade Items which are in a Particular Status

This script will export the trade items data into XML format which are in a particular status.

Overview of the script:

- Get the list of the trade items depending on a particular status, using the `searchTradeItem()` script operation and status as a search parameter.
- Generate an XML output for each trade item.

For a sample script refer [refer<link>](#)

Export Trade Items based on Local Status

This script will export the trade items data into XML format which are in a particular local status.

Overview of the script:

- Get the list of the trade items depending on a particular local status, using the `searchTradeItem()` script operation and status as a search parameter.
- Generate an XML output for each trade item.

Export Trade Items which belong to a Particular Category

This script will export the trade items data into XML format which belong to a particular category.

Overview of the script:

- Get the list of the trade items depending on a particular category, using the `searchTradeItem()` script operation with the category as the search parameter.
- Generate an XML output for each trade item.

Export Trade Items based on a search criteria

This script will export the trade items data into XML format based on a particular search criterion.

Overview of the script:

- Get the list of the trade items, using the searchTradeItem() script operation with search parameters. The list of search parameters are specified below.

The search parameters can be the following:

- GTIN,
 - GTIN_NAME,
 - BRAND_NAME
 - PRODUCT_NAME
 - DESCRIPTION
 - TARGET_MARKET
 - LANGUAGE_CODE
 - IP_GLN
 - IP_NAME
 - BRAND_OWNER_NAME
 - MANUFACTURER_GLN
 - MANUFACTURER_NAME
 - PRODUCT_TYPE
 - INTERNAL_CODE
 - INTERNAL_CODE_DESC
 - UDEX_CODE
 - UDEX_CATEGORY_DESC
 - GLOBAL_STATUS
 - UNIT_INDICATOR_YES(Value should be YES or NO)
 - UNIT_INDICATOR_NO(Value should be YES or NO)
 - UNIT_INDICATOR_NONE(Value should be YES or NO)
-
- Generate an XML output for each trade item.

Export All Trade Item Links

This script will export the trade item link data into XML format which belong to a particular category.

Overview of the script:

- Get the trade item using the getTradeItemSS() or searchTradeItem().
- Get the list of the trade item links for the trade item.
- Generate an XML output for each trade item link.

Export Trade Item Links Based on Status

This script will export the trade item link data into XML format which belong to a particular category.

Overview of the script:



- Get the list of Trade Item(GTIN) categories in the GTIN_Hierarchy with particular child item status.
- Construct the trade item link business objects using the GTIN categories.
- Generate an XML output for each trade item link.

Export Trading Partner

This script will export the trading partner data into XML format.

Overview of the script:

- Get the list of trading partners using the getTradingPartner (GLN).
- Generate an XML output for each trading partner.

Export Trading Partners – Search Based

This script will export the trading partners data into XML format based on the search criteria.

Overview of the script:

- Get the list of trading partners using the searchTradingPartners ().
 - GLN
 - PARTNER_NAME
 - STATUS
 - CONTACT
 - PARTY_ROLE
 - CITY_NAME
 - COUNTRY_CODE
 - LANGUAGE
- Generate an XML output for the trading partners.

Script Table

The following table lists the script operations, the arguments that each operation accepts and their descriptions. You can use a combination of these script operations to import/export data. The scripts listed are related to:

- [Trade Item](#)
- [Trade Item Links](#)
- [Trading Partner](#)
- [Notifications](#)
- [Transactions](#)



Script operations related to Trade Item

Trade Item			
Script Operation Name	Type	Prototype	Description
new\$TradeItemSS	constructor	TradeItem new TradeItemSS(String sGtin, String sGTINName, String sProductType, String sInternalCatCode, String sDataPoolCatCode)	Creates a new Supply Side Trade Item. sGTIN is the GTIN. sGTINName is the GTIN Name, sProductType is the Product type .
getTradeItemSS	method	TradeItem getTradeItemSS(String sGTIN,String sTM,String sIP)	Returns a Trade Item with the given GTIN, TM and IP values, else returns null.
setTradeItemAttributesSS	method	void TradeItem::setTradeItemAttributesSS(HashMap attributePathsToValues)	Sets the Trade Item with the given hashmap of attributepaths and their corresponding values. Note that multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
setTradeItemAttributeSS	method	void TradeItem::setTradeItemAttributeSS(String attributePath,String attributeValue)	Sets the attribute value for the attribute path of the trade item.
getTradeItemAttributeValueSS	static	String TradeItem::getTradeItemAttributeValueSS(String sAttrPath)	Returns the attribute value of the TradeItem.
validateGTIN	static	Boolean validateGTIN(String sGTIN)	Returns the attribute value of the TradeItem.
validateGLN	static	Boolean validateGLN(String sGLN)	Returns the attribute value of the TradeItem.
getCategory	static	ICategory getCategory(String sGTIN)	Returns the Category in the GTIN_Hierarchy.
getGTINTree	static	getGTINTree(String	Returns the GTIN when the GTIN String



Trade Item			
Script Operation Name	Type	Prototype	Description
		sGTIN)	is provided.
addChild	static	addChild(GTINTree tree)	add GTIN to the tree.
removeChild	static	removeChild(String sGTIN)	remove GTIN from the tree.
saveTradeItemSS	method	boolean TradeItem::saveTradeItemSS()	Saves the trade item to the database and returns true if the save was successful.
searchTradeItems	method	HashMap searchTradeItems(HashMap searchParamsToValues)	Searches and retrieves tradeItems based on a set of search parameters. The input is a hashmap of search keys and their values. The Keys should be one of the following and the case has to be upper case. GTIN, GTIN_NAME, BRAND_NAME, PRODUCT_NAME, DESCRIPTION, TARGET_MARKET, LANGUAGE_CODE, IP_GLN, IP_NAME, BRAND_OWNER_NAME, MANUFACTURER_GLN, MANUFACTURER_NAME, PRODUCT_TYPE, INTERNAL_CODE, INTERNAL_CODE_DESC, UDEX_CODE, UDEX_CATEGORY_DESC, GLOBAL_STATUS, UNIT_INDICATOR_YES (Value should be YES or NO), UNIT_INDICATOR_NO (Value should be YES or NO), UNIT_INDICATOR_NONE (Value should be YES or NO)
getTradeItemParentLinks	method	HashMap TradeItem::getTradeItemParentLinks()	Searches for parent trade item links for this trade item.
getTradeItemChildLinks	method	HashMap TradeItem::getTradeItemChildLinks()	Searches for child trade item links for this trade item.
getModifiedTradeItems	method	HashMap getModifiedTradeItems(String transactionType, Date startDate, Date endDate)	Searches and retrieves tradeItems modified between start date and end date for a given transaction type. Check the valid list of transactionType values in the Transaction_Type lookup table.
getItems	method	HashMap TradeItem::getItems()	Gets the Global and the local attributes for this TradeItem

Trade Item			
Script Operation Name	Type	Prototype	Description
setUDEXCategoryCode	method	void TradeItem::setUDEXCategoryCode(String code)	sets the trade item with the given udex category code Should be a valid UDEX Category code. Check the valid values in the UDEX_Hierarchy_UCCnet hierarchy.
setInternalCategoryCode	method	void TradeItem::setInternalCategoryCode(String code)	Sets the trade item with the given internal category code. Check the valid values in the Internal_Hierarchy_UCCnet hierarchy.
validateTradeItem	method	HashMap TradeItem::validateTradeItem()	This method validates the trade item. Returns HashMap only for the attributes which have failed validation.
validateTradeItemAttribute	method	String TradeItem::validateTradeItemAttribute(String sAttrPath)	This method validates the attribute for a trade item. Returns an error message if the attribute has errors else it returns null.
validateTradeItemAttributeGroup	method	HashMap TradeItem::validateTradeItemAttributeGroup(String sAttributeGroup)	This method validates an attribute group for a Trade Item. Returns HashMap only for the attributes which have failed validation.
getExchangeSpecificAttribute	static	HashMap getExchangeSpecificAttribute()	This method returns the exchange attributes which are a part of interoperability. Right now the implementation is specific to transora, not advised to use this for any other purpose
getAllTradeItemAttributes	method	HashMap TradeItem::getAllTradeItemAttributes()	Returns a hashmap of all tradeItem attributes to their values
deleteTMIPVariant	method	boolean TradeItem::deleteTMIPVariant()	Deletes the TMIP variant of this tradeItem.
deleteGTIN	method	boolean TradeItem::deleteGTIN()	Deletes the GTIN of this tradeItem. This can only be used when the tradeItem does not have any variants.
isTradeItemCheckedOut	static	boolean TradeItem::isTradeItemCheckedOut()	Returns whether the TradeItem is checked out in Workflow.
getWWREExchangeSpecificAttribute	method	HashMap TradeItem::getWWREExchangeSpecificAttribute()	This method returns the WWRE exchange attributes which are part of interoperability. Right now the

Trade Item			
Script Operation Name	Type	Prototype	Description
		bute()	implementation is specific to Transora, not advised to use this for any other purpose.
isWWREInteroperable	method	boolean TradeItem::isWWREInteroperable()	This method returns whether the trade item is WWRE Interoperable or not.
addItem_SS	method	void addItem_SS(HashMap entryMap, String strDate)	Adds the Approved TradeItem to Transora entryMap is an array containing TradeItem object . strDate is in "MM-dd-yyyy" format.
synchronizeItemChanges_SS	method	void synchronizeItemChanges_SS(HashMap entryMap, String strDate)	Synchronizes the TradeItem changes to Transora entryMap is an array containing TradeItem object . strDate is in "MM-dd-yyyy" format.
publishNewItem_SS	method	void publishNewItem_SS(HashMap entryMap, HashMap tradingPartners, String cascade, String strDate)	Publishes the Trade Item as New Item entryMap is an array containing TradeItem object . tradingPartners is an array containing trading partners GLNs. Cascade option is Yes or No. strDate is in "MM-dd-yyyy" format.
sendInitialLoad_SS	method	void sendInitialLoad_SS(HashMap entryMap, HashMap tradingPartners, String cascade, String strDate)	Publishes the Trade Item as Initial Load
addItemLinksForMassTransactions_SS	method	void addItemLinksForMassTransactions_SS(HashMap entryMap, String strDate)	Method to Add Item Links for Mass Transactions
getCtgItemAttributesValues	method	HashMap Item::getCtgItemAttributesValues()	Returns a HashMap mapping the paths (spec_name/attribute_name) of attributes to their respective values for all the attributes



Trade Item			
Script Operation Name	Type	Prototype	Description
readGDSProperties_SS	method	String readGDSProperties_SS(String key)	Method to read GDS Properties

Script operations related to Trade Item Link

Trade Item Link			
Script Operation Name	Type	Prototype	Description
new\$TradeItemLink	constructor	TradeItemLink new TradeItemLink(String parentGTIN, String childGTIN, [Integer linkQuantity])	Returns a new TradeItemLink with the parentGTIN and childGTIN. Optionally a link quantity can be passed. The default linkquantity is zero. Note that the for the tradeitem link to be created the parentGTIN and the childGTIN should exist in the system
setLinkAttributes	method	void TradeItemLink::setLinkAttributes(HashMap hmAttributeNamesToValues)	Sets the TradeItemLink with the given hashmap of attribute names and their corresponding values. Note that only attribute names need to be passed and not the full attribute path.
setLinkAttribute	method	void TradeItemLink::setLinkAttribute(String attributeName,String attributeValue)	Sets the attribute value for the attribute name of the TradeItemLink. Note that attributeName is just the attribute name and not the full path of the attribute.
getTradeItemLink	method	TradeItemLink getTradeItemLink(String parentGTIN, String childGTIN)	Returns a TradeItemLink between parentGTIN and the childGTIN.
saveTradeItemLink	method	boolean TradeItemLink::saveTradeItemLink()	Saves the TradeItemLink to the database and returns true if the save was successful.
getLinkAttributeValue	method	String TradeItemLink::getLinkAttributeValue(String	Returns the attribute value for the given attribute name in the Trade Item Link. Note that



Trade Item Link			
Script Operation Name	Type	Prototype	Description
		sAttrName)	sAttrName is only the attribute name and not the full path of the attribute

Script operations related to Trading Partner

Trading Partner			
Script Operation Name	Type	Prototype	Description
new\$TradingPartner	Constructor	TradingPartner new TradingPartner (String sGLN, String sPartnerName, String sCountry, String sGLNID)	Creates a new Trading Partner with the given GLN.
getTradingPartner	Static	getTradingPartner(String sGLN)	Returns a TradingPartner with the given GLN, else returns null.
setPartnerAttribute	Method	TradingPartner::setPartnerAttribute(String attributePath,String attributeValue)	Sets the attribute value for the attribute path of the Trading Partner.
setPartnerAttributes	Method	TradingPartner::setPartnerAttributes (HashMap hmAttributePathsToValues)	Sets the Trading Partner with the given hashmap of attributepaths and their corresponding values. Note: Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
saveTradingPartner	Method	TradingPartner::saveTradingPartner()	Saves the trading partner to the database and returns true if the save was successful.
searchTradingPartners	Static	TradingPartner::searchTradingPartners (HashMap hmSearchParamsToValues)	Searches and retrieves trading partners based on a set of search parameters. The input is a hashmap of search keys and their values. The Keys should be one of the following and the case has to be upper case. GLN PARTNER_NAME STATUS



Trading Partner			
Script Operation Name	Type	Prototype	Description
			CONTACT PARTY_ROLE CITY_NAME COUNTRY_CODE LANGUAGE
getPartnerAttributeValue	Method	TradingPartner::getPartnerAttributeValue (String sAttrPath)	Returns the attribute value for the given path in the trade item link.

Script operations related to Notifications

Notifications			
Script Operation Name	Type	Prototype	Description
createNotification	method	boolean createNotification(IBusinessObject iObj, String notifType, String userName)	Creates a notification object and returns true if the operation was successful. iObj is the Business Object. The business object can be a TradeItem or TradingPartner or an Item Link. notifType is the NotificationType. This can be retrieved from the NotificationSecurityMappingTable lookup table. Username is the user id of the user who is initiating the notification or who is responsible for the import/export of the item.

Script operations related to Transactions

Transactions			
Script Operation Name	Type	Prototype	Description
createTransaction	Static	createTransaction(IBusinessObject iObj, String txnType, String username)	<ul style="list-style-type: none"> Creates a transaction object and returns true if the operation was successful. iObj is the Business Object. The business object can be a



Transactions			
Script Operation Name	Type	Prototype	Description
			trade item or trading partner or and item link. <ul style="list-style-type: none">• txnType is the notification type. This can be retrieved from the NotificationSecurityMappingTable lookup table.• userName is the user ID of the user who is initiating the notification or the person who is responsible for the import/export of the item.

Using Script Operations

This section provides the sample implementations for trade items, trading partners, and trade item links.

Trade Items

The sample scripts for the following are provided:

- Create a New Trade Item, Create Transaction and Notification
- Global Enrichment of a Trade Item
- Getting a Trade Item
- Global and Local Enrichment of Trade Item
- Setting Attributes to a Trade Item
- Getting Attribute Values of a Trade Item
- Search Trade Items
- Get Modified Trade Items Between Dates
- Validating Trade Item
- Get Trade Items



Create a New Trade Item, Create Transaction, and Notification

The script operations used are new TradeItem, saveTradeItem, createTransaction, and createNotification.

```
var tradeItem = new TradeItemSS
("44444444444442", "GTIN_Name", "CASE", "10000056", "00000001.0000000
02.000000012");

tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GTINName
/languageCode", "en");

tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GTINName
/text", "GTIN_Name");

var itemSaved = tradeItem.saveTradeItemSS();
out.writeln("TradeItem [44444444444442] Saved ? : "+itemSaved);

var txnCreated =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mast
er");
out.writeln("Transaction Created ? : "+txnCreated);

var notfCreated =
createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mas
ter");
out.writeln("Notification sent ? : "+notfCreated);
```

Global Enrichment of a Trade Item

The script operations used are getTradeItem, setTradeItemAttribute, saveTradeItem, createTransaction, and createNotification.

```
var tradeItem = getTradeItemSS("44444444444442", "XX", "XX");
tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/Brand/Br
andName", "Coca Cola");
tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/Pack", "8
");
tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemS
tatus", "Global Enrichment Completed");

var itemSaved = tradeItem.saveTradeItemSS();
out.writeln("TradeItem [44444444444442] Saved ? : "+itemSaved);

var txnCreated =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "ma
ster");
out.writeln("Transaction Created ? : "+txnCreated);

var notfCreated =
createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "m
aster");
out.writeln("Notification sent ? : "+notfCreated);
```

Getting a Trade Item

```
var tradeItem = getTradeItemSS("44444444444442", "XX", "XX");
out.writeln(tradeItem);
```



Global and Local Enrichment of Trade Item

```
var tradeItem = new
TradeItemSS("45645666666660", "TESTING2", "CASE", "10000056", "0000000
01.000000002.000000012");

if(tradeItem != null)
{
    tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/B
rand/BrandName", "Coca Cola");
    tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/P
ack", "8");
    tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/G
DSItemStatus", "Draft With Variant");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/TargetMarket", "US");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/InformationProvider", "7981315112271");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/InformationProviderName", "TEST_GLN");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/IsPrivate", "true");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/DateInformation/StartAvailabilityDate", "+today().formatDate(
"dd-MMM-yyyy HH:mm"));
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/MeasureCharacteristics/Height/UOM", "IN");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/MeasureCharacteristics/Height/Value", "10");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/MeasureCharacteristics/Width/UOM", "IN");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/MeasureCharacteristics/Width/Value", "20");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/MeasureCharacteristics/Depth/UOM", "IN");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/MeasureCharacteristics/Depth/Value", "30");
    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/UnitIndicator/OrderingUnitIndicator", "true");

    tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_
Spec/GDSItemStatus", "Approved");

    var itemSaved = tradeItem.saveTradeItemSS();
    out.writeln("TradeItem [45645666666660] Saved ? :
"+itemSaved);

    var txnCreated =
createTransaction(tradeItem, "NEW_ITEM_APPROVED", "master");
    out.writeln("Transaction Created ? : "+txnCreated);

    var notifCreated =
createNotification(tradeItem, "NEW_ITEM_APPROVED", "master");
    out.writeln("Notification sent ? : "+notifCreated);
}
```

Setting Attributes to a Trade Item

The script operation used is `settradeitemattributesSS()`.

```
var tradeItem =
getTradeItemSS("45645666666660", "US", "7981315112271");
if(tradeItem != null)
```



```

{
    var hmAttrPathsToValues = [];

hmAttrPathsToValues["Global_Attributes_Spec/BrandDescription/languageCode"] = "en";

hmAttrPathsToValues["Global_Attributes_Spec/BrandDescription/text"] = "COKE";
    hmAttrPathsToValues["Global_Local_Attributes_Spec/InformationProviderName"] = "TEST IP";
    tradeItem.setTradeItemAttributesSS(hmAttrPathsToValues);
    var itemSaved = tradeItem.saveTradeItemSS();
    out.writeln("TradeItem [4564566666660] Saved ? :
"+itemSaved);
}

```

Getting Attribute Values of a Trade Item

```

var tradeItem =
getTradeItemSS("4564566666660","US","7981315112271");
if(tradeItem != null)
{
    var gtinName =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Attributes_Spec/GTINName/text"),"");
    var ip =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/InformationProvider"),"");

    out.writeln("GTINName is "+gtinName);
    out.writeln("IP is "+ip);
}

```

Search Trade Items

```

var hmTradeItems = [];
var hmSearchKeysToValues = [];

hmSearchKeysToValues["GTIN"] = "4564566666*";
hmSearchKeysToValues["UDEX_CODE"] =
"000000001.000000002.000000012*";

hmTradeItems = searchTradeItems(hmSearchKeysToValues);

if(hmTradeItems != null)
{
    out.writeln("Returned Trade Items are \n");
    for(i = 0; i < hmTradeItems.size(); i++)
    {
        out.writeln(hmTradeItems[i]);
    }
}
else
{
    out.writeln("No Trade Items found in Search");
}

```



Get Modified Trade Items between Dates

```
var txnType = "Global Attribute Enrichment";
var startDate = new Date("dd-MMM-yyyy", "27-AUG-2004");
var endDate = new Date("dd-MMM-yyyy", "31-AUG-2004");

var hmTradeItems = [];
hmTradeItems = getModifiedTradeItems(txnType, startDate, endDate);

if(hmTradeItems != null)
{
    out.writeln("Returned Trade Items are \n");
    for(i = 0; i < hmTradeItems.size(); i++)
    {
        out.writeln(hmTradeItems[i]);
    }
}
else
{
    out.writeln("No Trade Items found in Search");
}
```

Validating Trade Item

```
var tradeItem =
getTradeItemSS("45645666666660", "US", "7981315112271");
hmErrors = tradeItem.validateTradeItem();

out.writeln(hmErrors);
```

Get Trade Items

```
var tradeItem =
getTradeItemSS("45645666666660", "US", "7981315112271");
var items = tradeItem.getItems();

out.writeln(items);
```

Trading Partners

The scripts are provided for the following:

- Search Trading Partners
- Get Trading Partner Attribute Value
- Update an Existing Trading Partner
- Create a New Trading Partner

Create a New Trading Partner

```
var tradingPartner = new
TradingPartner("6701115112124", "TEST_GLN", "US", "Supplier");
tradingPartner.setPartnerAttribute("GLN_Spec/Status", "Approved");
var tpSaved = tradingPartner.saveTradingPartner();

out.writeln("TradingPartner [6701115112124] Saved ? : "+tpSaved);

var txnCreated =
createTransaction(tradingPartner, "NEW_TRADING_PARTNER_APPROVED", "m
aster");
```



```
out.writeln("Transaction Created ? : "+txnCreated);

var notifCreated =
createNotification(tradingPartner,"NEW_TRADING_PARTNER_APPROVED",
"master");
out.writeln("Notification sent ? : "+notifCreated);
```

Update an Existing Trading Partner

```
var tradingPartner = getTradingPartner("6701115112124");
var hmAttrPathsToValues = [];
hmAttrPathsToValues["GLN_Spec/Trading Partner State"] = "CA";
tradingPartner.setPartnerAttributes(hmAttrPathsToValues);

var tpSaved = tradingPartner.saveTradingPartner();
out.writeln("TradingPartner [6701115112124] Saved ? : "+tpSaved);

var txnCreated =
createTransaction(tradingPartner,"MODIFY_TRADING_PARTNER_APPROVED",
"master");
out.writeln("Transaction Created ? : "+txnCreated);

var notifCreated =
createNotification(tradingPartner,"MODIFY_TRADING_PARTNER_APPROVED",
"master");
out.writeln("Notification sent ? : "+notifCreated);
```

Get Trading Partner Attribute Value

```
var tradingPartner = getTradingPartner("6701115112124");
var tpLanguage =
tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner
Language");
var tpID = tradingPartner.getPartnerAttributeValue("GLN_Spec/GLN
Identifier");
var tpState =
tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner
State");
out.writeln("Trading Partner language is "+tpLanguage);
out.writeln("Trading Partner ID is "+tpID);
out.writeln("Trading Partner State is "+tpState);
```

Search Trading Partners

```
var hmSearchKeysToValues = [];
hmSearchKeysToValues["PARTNER_NAME"] = "TEST_GLN";
var hmTradingPartners =
searchTradingPartners(hmSearchKeysToValues);

if(hmTradingPartners != null)
{
    out.writeln("Returned Trading Partners are \n");
    for(i = 0; i < hmTradingPartners.size(); i++)
    {
        out.writeln(hmTradingPartners[i]);
    }
}
else
{
    out.writeln("No hmTradingPartners found in Search");
}
```

Trade Item Links

The scripts are provided for the following:

- Create a New Trade Item Link
- Update Trade Item Links
- Get Trade Item Links Attributes
- Get Child Links
- Get Parent Links

Create a New Trade Item Link

```
var tradeItem = new
TradeItemSS("45645666666677", "CHILD", "EACH", "10000056", "000000001.
000000002.000000012");

var itemSaved = tradeItem.saveTradeItemSS();
out.writeln("TradeItem [45645666666677] Saved ? : "+itemSaved);

var txnCreated =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mast
er");
out.writeln("Transaction Created ? : "+txnCreated);

var notfCreated =
createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mas
ter");
out.writeln("Notification sent ? : "+notfCreated);

var tradeItemLink = new
TradeItemLink("45645666666660", "45645666666677", 1);

var linkSaved = tradeItemLink.saveTradeItemLink();
out.writeln("45645666666660 < == > 45645666666677 [LINK] Saved ?
:"+linkSaved);
var bCreateTxn =
createTransaction(tradeItemLink, "MODIFY_ITEM_LINK_APPROVAL_PENDING
", "master");
out.writeln("Transaction Created ? : "+bCreateTxn);
var bCreateNotf =
createNotification(tradeItemLink, "MODIFY_ITEM_LINK_APPROVAL_PENDIN
G", "master");
out.writeln("Notification sent ? : "+bCreateNotf);
```

Update Trade Item Links

```
var tradeItemLink =
getTradeItemLink("45645666666660", "45645666666677");
tradeItemLink.setLinkAttribute("Status", "Approved");
var hmAttrsToValues = [];
hmAttrsToValues["Quantity"] = "2";
tradeItemLink.setLinkAttributes(hmAttrsToValues);
var linkSaved = tradeItemLink.saveTradeItemLink();
out.writeln("45645666666660 < == > 45645666666677 [LINK] Saved ?
:"+linkSaved);
```

```

var bCreateTxn =
createTransaction(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING
","master");
out.writeln("Transaction Created ? : "+bCreateTxn);
var bCreateNotf =
createNotification(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING",
"master");
out.writeln("Notification sent ? : "+bCreateNotf);

```

Get Trade Item Links Attributes

```

var tradeItemLink =
getTradeItemLink("45645666666660","45645666666677");

var qty = tradeItemLink.getLinkAttributeValue("Quantity");
var status = tradeItemLink.getLinkAttributeValue("Status");
var childGtin =
tradeItemLink.getLinkAttributeValue("GlobalTradeItemNumber");

out.writeln("Quantity is "+qty);
out.writeln("Status is "+status);
out.writeln("ChildGtin is "+childGtin);

```

Get Child Links

```

var tradeItem =
getTradeItemSS("45645666666660","US","7981315112271");

var hmChildLinks = [];
hmChildLinks = tradeItem.getTradeItemChildLinks();

if(hmChildLinks != null)
{
    out.writeln("Returned Child Trade Item Links are \n");
    for(i = 0; i < hmChildLinks.size(); i++)
    {
        out.writeln(hmChildLinks[i]);
    }
}
else
{
    out.writeln("No Child TradeItem Links found in Search");
}

```

Get Parent Links

```

var tradeItem = getTradeItemSS("45645666666677","XX","XX");

var hmParentLinks = [];
hmParentLinks = tradeItem.getTradeItemParentLinks();

if(hmParentLinks != null)
{
    out.writeln("Returned Parent Trade Item Links are \n");
    for(i = 0; i < hmParentLinks.size(); i++)
    {
        out.writeln(hmParentLinks[i]);
    }
}
else

```



```
{  
    out.writeln("No Parent TradeItem Links found in Search");  
}
```

Spec Maps

```
specMap = getSpecMapByName("GDS XML GLN Attributes Spec Map");  
var hmDstPathsToSrcPaths = specMap.getMappings();  
  
out.writeln(hmDstPathsToSrcPaths);
```

Ch 5 Import/Export –Demand Side

The Import/Export operation allows the user to export or import information from or to IBM Global Data Synchronization for WebSphere Product Center. The user can import or export information about trade items, trade item links, and trading partners. This chapter provides a few guidelines for the PSS personnel while implementing IBM Global Data Synchronization for WebSphere Product Center for the customer. The various scenarios for import/export are explained.

Note: IBM Global Data Synchronization for WebSphere Product Center does not have separate import/export module. For release 1.2, WebSphere Product Center 5.1.1.1 is used for import/export.

The import or export depends a lot on the data available at the customer location. Some may not have GTIN as the primary key for trade items, while for some the mandatory fields may be missing. In such cases, the PSS team should setup the import or export process correctly.

If the GTIN is not present in legacy systems, a GTIN generator needs to be plugged in while importing the data from legacy systems. The generated GTIN need not be stored in the legacy systems and can be dynamically generated. Ensure that the generated GTIN is unique. It should be the same every time the data is imported or exported and not be a randomly generated number.

If the source data does not contain mandatory fields (required by IBM Global Data Synchronization for WebSphere Product Center), a dummy value needs to be generated. The trade items can be modified after the import.

The trade items can be imported in any desired status. This can be done while performing import or export by modifying the import/export script.

File Formats for Import/Export

The data can be imported or exported in different file formats. As a sample, imports or exports is provided in CSV (Comma Separated Values) and XML formats. This can be extended to suite the customer's requirements.

Implementation Approach for CSV

A trade item is represented by two specs:

- Global_Attributes_Spec
- Global_Local_Attributes_Spec

When setting up an export operation, a destination spec needs to be defined that maps to these two specs. The destination spec maps the attribute names of the trade items in the legacy systems to the attribute names of trade items in IBM Global Data Synchronization for Websphere Product Center.



Similarly, when setting up import, a file spec needs to be defined that maps to these two specs. The file spec maps the attribute names of the trade items in IBM Global Data Synchronization for Websphere Product Center to the attribute names of trade items in the legacy systems.

Implementation Approach for XML

Using the GDS Business Object script operation, the xml/csv file can be imported into IBM Global Data Synchronization for WebSphere Product Center.

Scenarios

This section lists down the various customer scenarios for import and export, using the various script operations provided by both IBM Global Data synchronization for WebSphere Product Center and WebSphere Product Center.

Note: For information o WebSphere Product Center script operations, refer the *WebSphere Product Center User's Guide*.

Note: For more information on individual script operations in IBM Global Data synchronization for WebSphere Product Center, refer to the section [using script operations](#)

For Import

This section explains the various scenarios and gives an overview of the import scripts.

Note: When an item is imported through scripts, unlike the GDS UI, it does not go through an approval process. All approval processes are implicit and the user has to set the appropriate final status for the trade item. For the list of status changes refer to the [trade item status list](#) table.

- [Import Global Trade Items](#)
- [Import Global and Local Trade Items](#)
- [Import Trade Item in a Particular Status](#)
- [Update an Already Existing Trade Item through Import](#)
- [Import Trade Item Links in a Particular Status](#)
- [Updating Trade Item Links through Import](#)
- [Import Trading Partners](#)
- [Updating a trading partners through import](#)

Import Global Trade Items

For this scenario a specmap is created in the system which maps the xml attributes to the Global_Attributes_Spec, FMCG_Global_Attributes_Spec, Hardlines_Global_Attributes_Spec and Office_Supplies_Global_Attributes_Spec of the Global_Catalog. Then use the script function specMap.getMappings () which returns a hashmap of the destination paths to the source paths. Using this, the xml nodes are parsed and the corresponding attribute path values are set.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the Global_Attributes_Spec, FMCG_Global_Attributes_Spec, Hardlines_Global_Attributes_Spec and Office_Supplies_Global_Attributes_Spec of the Global_Catalog. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec map.
- Create a trade item business object with the required extension.
- Set the obtained attribute values on the trade item business object.
- Set the appropriate status for the global item in the trade item business object.
- Save the trade item business object.
- Create a transaction log for the imported trade item.
- Notify the user by creating an appropriate notification.(optional)

Global Item status	Local Item status
Draft	Not applicable
Global enrichment completed	

Import Global and Local Trade Items

This script will import the trade item with a variant, consisting of global item and global local item into the Global_catalog and Global_local_catalog respectively.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the Global_Attributes_Spec, FMCG_Global_Attributes_Spec, Hardlines_Global_Attributes_Spec and Office_Supplies_Global_Attributes_Spec of the Global_Catalog and Global_Local_Attributes_Spec FMCG_Global_Local_Attributes_Spec, Hardlines_Global_Local_Attributes_Spec and



Office_Supplies_Global_Local_Attributes_Spec of the Global_Local_Catalog.
The spec map is created for the data to be imported.

- Obtain the attribute values using the spec maps.
- Create a trade item business object with the required extension.
- Set the obtained attribute values on the trade item business object.
- Set the appropriate status for the global item and the local item in the trade item business object.
- Depending on the current status of the trade item, ensure that the trade item passes the compliance check.
- Save the trade item business object.
- Create a transaction log for the imported trade item.
- Notify the user by creating an appropriate notification.(optional)

Update an Already Existing Trade Item through Import

This script will import the trade item with a variant, consisting of global item and global local item into the Global_catalog and Global_local_catalog respectively.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the Global_Attributes_Spec, FMCG_Global_Attributes_Spec, Hardlines_Global_Attributes_Spec and Office_Supplies_Global_Attributes_Spec of the Global_Catalog and Global_Local_Attributes_Spec FMCG_Global_Local_Attributes_Spec, Hardlines_Global_Local_Attributes_Spec and Office_Supplies_Global_Local_Attributes_Spec of the Global_Local_Catalog.
The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Get the handle of the trade item business object.
- Set the obtained attribute values on the trade item business object.
- Set the appropriate status for the global item and the local item in the trade item business object.
- Depending on the current status of the trade item, ensure that the trade item passes the compliance check.
- Save the trade item business object.
- Create a transaction log for the imported trade item.
- Notify the user by creating an appropriate notification.(optional)



Trade Item Status

The following table shows the rules that govern status of trade items.

For Global Item:

From Status	To Status	Compliance Check Required	Notification	Transaction
Draft	Draft	NO		0
Draft With Variant	Draft With Variant	NO		
Global Attributes Modified Locally	Global Attributes Modified Locally	YES		0
Global Enrichment Completed	Global Enrichment Completed	NO		
Pending Variant Creation	Pending Variant Creation	NO		
Modified Compliant	Modified Compliant	YES		
Modified Compliant Approval Pending	Modified Compliant Approval Pending	YES		
Registered	Modified Locally	YES	MODIFY_ITEM_APPROVED	MODIFY_ITEM_APPROVED

For Local Item

From Status	To Status	Compliance Check Required	Notification	Transaction
Draft	Draft	NO		0
Draft With Variant	Draft With Variant	NO		
Global Attributes Modified Locally	Global Attributes Modified Locally	YES		0
Global Enrichment	Global Enrichment	NO		

From Status	To Status	Compliance Check Required	Notification	Transaction
Completed	Completed			
Pending Variant Creation	Pending Variant Creation	NO		
Modified Compliant	Modified Compliant	YES		
Modified Compliant Approval Pending	Modified Compliant Approval Pending	YES		
Modified Locally	Modified Compliant Approval Pending	YES	MODIFY_ITEM_APPROVAL_PENDING	MODIFY_ITEM_APPROVAL_PENDING
Complete	Edited	YES		0
Compliant	Edited	YES		0
Compliant	Compliant	YES		
Approved	Edited	NO		0
Rejected	Compliant	YES		
Registered	Modified Locally	YES	MODIFY_ITEM_APPROVED	MODIFY_ITEM_APPROVED

Import Trade Item Links

This script will import the trade item links into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the GTIN_Hierarchy_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Create a trade item link business object.
- The link should have valid combinations of the product types for the parent and the child GTIN.
- Set the obtained attribute values on the trade item link business object.
- Set the appropriate status for the trade item link business object.
- Save the trade item link business object.



- Create a transaction log for the imported trade item link.
- Notify the user by creating an appropriate notification.(optional)

Update Trade Item Links through Import

This script will update the trade item links into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes to the GTIN_Hierarchy_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Obtain a trade item link business object.
- The link should have valid combinations of the product types for the parent and the child GTIN.
- Set the obtained attribute values on the trade item link business object.
- Set the appropriate status for the trade item link business object.
- Save the trade item link business object.
- Create a transaction log for the imported trade item link.
- Notify the user by creating an appropriate notification.(optional)

The following table shows the rules that govern the status of trade item links.

From_Status	To_Status	Notification	Transaction
Draft	Draft		
Approved	Edited	MODIFIED_ITEM_LINK_APPROVAL_PENDING	MODIFIED_ITEM_LINK_APPROVAL_PENDING
Rejected	Edited	MODIFIED_ITEM_LINK_APPROVAL_PENDING	MODIFIED_ITEM_LINK_APPROVAL_PENDING
Registered	Modified Locally	MODIFIED_ITEM_LINK_APPROVED	MODIFIED_ITEM_LINK_APPROVED
Modified Locally	Modified Locally		

Import Trading Partners

This script will import the trading partners into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes GLN_Spec. The spec map is created for the data to be imported.



- Obtain the attribute values using the spec maps.
- Create a trading partner business object.
- Set the attribute “item type” as “Trading Partner” on the trading partner business object.
- Set the obtained attribute values on the trading partner business object.
- Set the appropriate status for the trading partner business object.
- Save the trading partner business object.
- Create a transaction log for the imported trading partner.
- Notify the user by creating an appropriate notification.(optional)

Update Trading Partners through import

This script will import the trading partners into the IBM Global Data Synchronization for Websphere Product Center.

Overview of the import script:

- Create a spec Map which maps the XML attributes GLN_Spec. The spec map is created for the data to be imported.
- Obtain the attribute values using the spec maps.
- Obtain a trading partner business object.
- Set the obtained attribute values on the trading partner business object.
- Set the appropriate status for the trading partner business object.
- Save the trading partner business object.
- Create a transaction log for the imported trading partner.
- Notify the user by creating an appropriate notification.(optional)

The following table shows the rules that govern auto status of trading partners.

Existing Status	After Import	Notification	Transaction
Draft	Draft		0
Approved	Edited		0
Rejected	Edited		0
Edited	Edited		0

Handling Validations

Three validation scripts have been exposed to check if the trade item is compliant with an exchange. They are.

4. HashMap TradeItem::validateTradeItem()

This method validates all the attributes for a TradeItem object and returns a HashMap of error messages only for those attributes which have failed validation.

5. String TradeItem::validateTradeItemAttribute(String sAttrPath)

This method validates the attribute for a TradeItem object and returns an error message if the attribute has errors else it returns null.

6. HashMap

TradeItem::validateTradeItemAttributeGroup(StringsAttributeGroup)

This method validates an attribute group for a TradeItem object and returns a HashMap of error messages only for those attributes which have failed validation.

Based on the business requirement these functions can be called in the import script before saving the trade item.

Export

Note: All the Export scripts refer to a library script called GDSLibrary.GenerateXML. This script basically generates the xml definition for trade item, trading partners, and trade item links. This script is as shown.

The various scenarios for export are:

- [Export trade items that have been created or updated in a date range](#)
- [Export Trade Items which are in a Particular Status](#)
- [Export Trade Items Based on Local Status](#)
- [Export Trade Items Based on Particular Category](#)
- [Export Trade Items Based on a search criteria](#)
- [Export All Trade Item Links](#)
- [Export Trade Item Links Based in status](#)
- [Export trading Partner](#)
- [Export Trading Partner – Search Based](#)

Export Trade Items that have been created or updated in a date range

This script will export the trade items data into XML format that has been created or updated between a certain date range.

Overview of the script:

- Get the list of the trade items depending on a particular transaction type using the getModifiedTradeItems() script operation

- Generate an XML output for each trade item.

Export Trade Items which are in a Particular Status

This script will export the trade items data into XML format which are in a particular status.

Overview of the script:

- Get the list of the trade items depending on a particular status, using the searchTradeItem() script operation and status as a search parameter.
- Generate an XML output for each trade item.

Export Trade Items based on Local Status

This script will export the trade items data into XML format which are in a particular local status.

Overview of the script:

- Get the list of the trade items depending on a particular local status, using the searchTradeItem() script operation and status as a search parameter.
- Generate an XML output for each trade item.

Export Trade Items which belong to a Particular Category

This script will export the trade items data into XML format which belong to a particular category.

Overview of the script:

- Get the list of the trade items depending on a particular category, using the searchTradeItem() script operation with the category as the search parameter.
- Generate an XML output for each trade item.

Export Trade Items based on a search criteria

This script will export the trade items data into XML format based on a particular search criteria.

Overview of the script:

- Get the list of the trade items, using the searchTradeItem() script operation with search parameters. The list of search parameters are specified below.

The search parameters can be the following:

- GTIN,
- GTIN_NAME,
- BRAND_NAME
- PRODUCT_NAME
- DESCRIPTION



- TARGET_MARKET
 - LANGUAGE_CODE
 - IP_GLN
 - IP_NAME
 - BRAND_OWNER_NAME
 - MANUFACTURER_GLN
 - MANUFACTURER_NAME
 - PRODUCT_TYPE
 - INTERNAL_CODE
 - INTERNAL_CODE_DESC
 - UDEX_CODE
 - UDEX_CATEGORY_DESC
 - GLOBAL_STATUS
 - UNIT_INDICATOR_YES(Value should be YES or NO)
 - UNIT_INDICATOR_NO(Value should be YES or NO)
 - UNIT_INDICATOR_NONE(Value should be YES or NO)
-
- Generate an XML output for each trade item.

Export All Trade Item Links

This script will export the trade item link data into XML format which belong to a particular category.

Overview of the script:

- Get the trade item using the `getTradeItemSS()` or `searchTradeItem()`.
- Get the list of the trade item links for the trade item.
- Generate an XML output for each trade item link.

Export Trade Item Links Based on Status

This script will export the trade item link data into XML format which belong to a particular category.

Overview of the script:

- Get the list of Trade Item(GTIN) categories in the `GTIN_Hierarchy` with particular child item status.
- Construct the trade item link business objects using the GTIN categories.
- Generate an XML output for each trade item link.

For a sample script refer<link>

Export Trading Partner

This script will export the trading partner data into XML format.



Overview of the script:

- Get the list of trading partners using the getTradingPartner (GLN).
- Generate an XML output for each trading partner.

Export Trading Partners – Search Based

This script will export the trading partners data into XML format based on the search criteria.

Overview of the script:

- Get the list of trading partners using the searchTradingPartners ().
 - GLN
 - PARTNER_NAME
 - STATUS
 - CONTACT
 - PARTY_ROLE
 - CITY_NAME
 - COUNTRY_CODE
 - LANGUAGE
- Generate an XML output for the trading partners.

Script Table

The following table lists the script operations, the arguments that each operation accepts and their descriptions. You can use a combination of these script operations to import/export data. Listed below are script operation related to

- [Trade Item](#)
- [Trading Partner](#)
- [Notifications](#)
- [Transactions](#)
- [Messaging](#)
- [Spec Map](#)

Script operations related to Trade Item

Script Operation Name	Type	Prototype	Description
new\$TradeItemDS	constructor	TradeItem new TradeItemDS(String sEANUCCCode, String sEANUCCCodeType, String	Creates a new TradeItem with the given EAN UCC Code, EAN UCC Type, Core Extensions, Product Type and maps it to sInternalCatCode. Optionally the GTIN, GTIN Name and the UDEX category code can be supplied

Script Operation Name	Type	Prototype	Description
		sExtensionName, String sProductType,String sInternalCatCode, [[String sGTIN],[String sGTINName],[String sDataPoolCatCode]	can be supplied
getExchange SpecificAttribute	static	HashMap getExchangeSpecificAttribute()	This method returns the exchange attributes which are part of interoperability. Right now the implementation is specific to transora,not advised to use this for any other purpose
getTradeItemDS	method	TradeItem getTradeItemDS(String sEANUCCCode,String sTM,String sIP,[String sGTIN])	Returns a Trade Item with the given sEANUCCCode, sTM, sIP values and optionally sGTIN, else returns null.
setTradeItem AttributesDS	method	void TradeItem::setTradeItemAttributesDS(HashMap hmAttributePathsToValues)	Sets the Trade Item with the given hashmap of attributepaths and their corresponding values.Note that multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
setTradeItem AttributeDS	method	void TradeItem::setTradeItemAttributeDS(String attributePath,String attributeValue)	Sets the attribute value for the attribute path of the trade item.
getTradeItem AttributeValueDS	static	String TradeItem::getTradeItemAttributeValueDS(String sAttrPath)	Returns the attribute value of the TradeItem.
saveTradeItemDS	method	boolean TradeItem::saveTradeItemDS()	Saves the trade item to the database and returns true if the save was successfull.
getAllTradeItemAttributes	method	HashMap TradeItem::getAllTradeItemAttributes()	Returns a hashmap of all tradeItem attributes to their values
searchTradeItemsDS	method	HashMap searchTradeItemsDS(HashMap hmSearchParamsToValues)	Searches and retrieves tradeItems based on a set of search parameters. The input is a hashmap of search keys and their values. The Keys should be one of the following and the case has to be upper



Script Operation Name	Type	Prototype	Description
			case.EAN_UCC_CODE,GTIN, GTIN_NAME, BRAND_NAME, PRODUCT_NAME, DESCRIPTION, TARGET_MARKET, LANGUAGE_CODE, IP_GLN, IP_NAME, BRAND_OWNER_NAME, MANUFACTURER_GLN,MANUFACTURER_NAME, PRODUCT_TYPE, UDEX_CODE, UDEX_CATEGORY_DESC,GLOBAL_STATUS, UNIT_INDICATOR_YES(Value should be YES or NO), UNIT_INDICATOR_NO(Value should be YES or NO), UNIT_INDICATOR_NONE(Value should be YES or NO)
deleteTMIPVariant	method	boolean TradeItem::deleteTMIPVariant()	Deletes the TMIP variant of this tradeItem.
deleteGTIN	method	boolean TradeItem::deleteGTIN()	Deletes the GTIN of this tradeItem. This can only be used when the tradeItem does not have any variants.
isTradeItemCheckedOut	static	boolean TradeItem::isTradeItemCheckedOut()	Returns whether the TradeItem is checked out in Workflow.
validateGTIN	static	Boolean validateGTIN(String sGTIN)	Returns the attribute value of the TradeItem.
validateGLN	static	Boolean validateGLN(String sGLN)	Returns the attribute value of the TradeItem.
getCategory	static	ICategory getCategory(String sGTIN)	Returns the Category in the GTIN_Hierarchy.
getGTINTree	static	getGTINTree(String sGTIN)	Returns the GTIN when the GTIN String is provided.
addChild	static	addChild(GTINTree tree)	add GTIN to the tree.
removeChild	static	removeChild(String sGTIN)	remove GTIN from the tree.
getTradeItemChildLinks	method	HashMap TradeItem::getTradeItemChildLinks()	Searches for child trade item links for this trade item.



Script Operation Name	Type	Prototype	Description
getModifiedTradeItems	method	HashMap getModifiedTradeItems(String transactionType, Date startDate, Date endDate)	Searches and retrieves tradeItems modified between start date and end date for a given transaction type. Check the valid list of transactionType values in the Transaction_Type lookup table.
getItems	method	HashMap TradeItem::getItems()	Gets the Global and the local attributes for this TradeItem
setUDEXCategoryCode	method	void TradeItem::setUDEXCategoryCode(String code)	sets the trade item with the given udex category code Should be a valid UDEX Category code. Check the valid values in the UDEX_Hierarchy_UCCnet hierarchy.
setInternalCategoryCode	method	void TradeItem::setInternalCategoryCode(String code)	Sets the trade item with the given internal category code. Check the valid values in the Internal_Hierarchy_UCCnet hierarchy.

Script operations related to Trade Item Links

Script Operation Name	Prototype	Description
new\$TradeItemLink	String parentGTIN, String childGTIN, [Integer linkQuantity]	Returns a new trade item link with the parent GTIN and child GTIN. Optionally a link quantity can be passed. The default link quantity is zero. Note: For the trade item link to be created the parent GTIN and the child GTIN should exist in the system.
getTradeItemLink	String parentGTIN, String childGTIN	Returns a trade item link between parent GTIN and the child GTIN.
TradeItemLink.setLinkAttribute	String attributePath, String attributeValue	Sets the attribute value for the attribute path of the trade item link.
TradeItemLink.setLinkAttributes	HashMap hmAttributePathsToValues	Sets the trade item link with the given hashmap of attribute paths and their corresponding values. Note: Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
TradeItemLink.saveTradeItemLink	NONE	Saves the trade item link to the database and returns true if the save was successful.
TradeItemLinks:getLinkAttribute	String sAttrPath	Returns the attribute value for the given



buteValue		path in the Trade Item Link
getModifiedTradeItemLinks	TBD	TBD

Script operations related to Trading partner

Trading Partner			
Script Operation Name	Type	Prototype	Description
new\$TradingPartner	Constructor	TradingPartner new TradingPartner (String sGLN, String sPartnerName, String sCountry, String sGLNID)	Creates a new Trading Partner with the given GLN.
getTradingPartner	Static	getTradingPartner(String sGLN)	Returns a TradingPartner with the given GLN, else returns null.
setPartnerAttribute	Method	TradingPartner::setPartnerAttribute(String attributePath,String attributeValue)	Sets the attribute value for the attribute path of the Trading Partner.
setPartnerAttributes	Method	TradingPartner::setPartnerAttributes (HashMap hmAttributePathsToValues)	Sets the Trading Partner with the given hashmap of attributepaths and their corresponding values. Note: Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
saveTradingPartner	Method	TradingPartner::saveTradingPartner()	Saves the trading partner to the database and returns true if the save was successful.
searchTradingPartners	Static	TradingPartner::searchTradingPartners (HashMap hmSearchParamsToValues)	Searches and retrieves trading partners based on a set of search parameters. The input is a hashmap of search keys and their values. The Keys should be one of the following and the case has to be upper case. GLN PARTNER_NAME STATUS CONTACT PARTY_ROLE CITY_NAME COUNTRY_CODE LANGUAGE



Trading Partner			
Script Operation Name	Type	Prototype	Description
getPartnerAttributeValue	Method	TradingPartner::getPartnerAttributeValue (String sAttrPath)	Returns the attribute value for the given path in the trade item link.

Script operations related to Notifications

Notifications			
Script Operation Name	Type	Prototype	Description
createNotification	method	boolean createNotification(IBusinessObject iObj, String notifType, String userName)	Creates a notification object and returns true if the operation was successful. iObj is the Business Object. The business object can be a TradeItem or TradingPartner or an Item Link. notifType is the NotificationType. This can be retrieved from the NotificationSecurityMappingTable lookup table. UserName is the user id of the user who is initiating the notification or who is responsible for the import/export of the item.

Script operations related to Transactions

Transactions			
Script Operation Name	Type	Prototype	Description
createTransaction	Static	createTransaction(IBusinessObject iObj, String txnType, String username)	<ul style="list-style-type: none"> Creates a transaction object and returns true if the operation was successful. iObj is the Business Object. The business object can be a trade item or trading partner or and item link. txnType is the notification type. This can be retrieved from the NotificationSecurityMappingTable lookup table. userName is the user ID of the user who is initiating the notification or the person who is responsible for



Transactions			
Script Operation Name	Type	Prototype	Description
			the import/export of the item.

Script operations related to Messaging

Messaging		
Script Operation Name	Usage	Description
New\$MessageItemHandler (String correlationId)	<pre>msgId = "3"; msgItemHandler = new MessageItemHandler(msgId);</pre>	<p>This operation can be used to instantiate a new MessageItemHandler for the specified correlation Id. The correlationId is the primary key in the MessageArchiveCatalog. The MessageItemHandler exposes methods to retrieve the Trade Items and link information from the CIN.</p>
HashMap MessageItemHandler :: getTradeItems()	<pre>msgId = "3"; msgItemHandler = new MessageItemHandler(msgId); items = msgItemHandler.getTradeItems(); out.writeln("Items are " + items); out.writeln("GTIN is " + items["0"].getTradeItemAttributeVa lueDS("Global_Attributes_Spec/Glob alTradeItemNumber"));</pre>	<p>The getTradeItems() method can be invoked on the MessageItemHandler object. This returns a HashMap of ITradeItems present in the message item represented by the MessageItemHandler.</p> <p>Note: For List of API's exposed on the TradeItem (like the getTradeItemAttribute ValueDS used above) refer GDS Documentation.</p>
HashMap MessageItemHandler ::	<pre>msgId = "3"; msgItemHandler = new</pre>	<p>The getLinks() method can be invoked on the</p>

Messaging		
Script Operation Name	Usage	Description
MessageItemHandler :: getLinks()	<pre>MessageItemHandler(msgId); links = msgItemHandler.getLinks() out.writeln("Items are:" + items["0"].getLinkAttributeValue("GlobalTradeItemNumber")) ;</pre>	<p>MessageItemHandler object. This returns a HashMap of TradeItemLink objects present in the message item represented by the MessageItemhandler.</p> <p>Note: For List of API's exposed on the TradeItemLink (like the getLinkAttributeValue used above) refer GDS Documentation.</p>
HashMap getNewPublicationMessageIds()	<pre>items = getNewPublicationMessageIds(); out.writeln("First Id is " + items["0"]);</pre>	<p>This operation can be used to search all the CIN's that have been received by GDS. Only those message item ids that have not been read by the external system and are not checked out in any workflows will be returned (i.e. where the external read status is "NEW"). The ids are returned as a HashMap</p>
HashMap getNewPublicationMessageIdsByType(HashMap CINTypes)	<pre>arr = []; arr[0] = "NEW_ITEM"; arr[1] = "INITIAL_LOAD"; items = getNewPublicationMessageIdsByType(arr); out.writeln("First Value is " + items["0"]);</pre>	<p>This operations returns the message item ids (each message item represents a CIN) based upon the types specified. Following are the valid types:</p> <p><u>Types for Uccnet:</u></p> <p>"NEW_ITEM", "INITIAL_LOAD", "ITEM_CHANGE","ITEM_DELIST","ITEM_WITHDRAW";</p> <p><u>Types for WWRE:</u></p> <p>"NEW_ITEM", "INITIAL_LOAD",</p>



Messaging		
Script Operation Name	Usage	Description
		<p>“ITEM_CHANGE_BY_REFRESH”, “ITEM_CORRECT”</p> <p>The operation returns a HashMap of the all the “NEW” message item ids of all the requested types. Items that have already been “READ” or that are checked out in any workflow are not returned.</p>
HashMap getPublicationMessageIdsByDate(String fromDate, String toDate)	<pre> toDate = "28-02-2005 00:00:00"; fromDate = "10-01-2005 19:20:00"; items = getPublicationMessageIdsByDate(fromDate, toDate); out.writeln("First MsgItem Id is " + items["0"]); </pre>	<p>This operation returns all the message item ids for CIN’s of all different types between the dates specified. All items, irrespective of the external read status i.e. both “READ” and “NEW”, are returned. Items that are checked out in any workflow will not be returned.</p> <p>Note: The dates should be specified in the following format “dd-MM-yyyy HH:mm:ss” [Example : “28-06-2005 10:15:30”]</p>
Boolean MessageItemHandler :: changeState()	<pre> msgId = "3"; msgItemHandler = new MessageItemHandler(msgId); result = msgItemHandler.changeState(); out.writeln("Is status changed " + result); </pre>	<p>This operation is invoked on the MessageItemHandler object to change the external read status of the message item (represented by the MessageItemHandler) to “READ”.</p> <p>This indicates that the external system has read this particular CIN and further searches based on type or new ids will not return this message item.</p> <p>It should be noted that the external system needs to invoke this method once the</p>

Messaging		
Script Operation Name	Usage	Description
		<p>system has processed the CIN.</p> <p>This method returns a Boolean value.</p>
HashMap sendConfirmationMessage (String confirmationType, HashMap correlationID)	<pre>ids = []; ids.add("3"); hmMap = sendConfirmation("ACCEPT_MSG",ids) ; out.writeln("Return is " + hmMap);</pre>	<p>This operation is invoked to send a confirmation message to the data pool. The type of message that is to be sent and the correlation ids of the message Items are specified as the parameters.</p> <p>Following is the list of message types that can be sent:</p> <ul style="list-style-type: none"> • REVIEW_MSG • ACCEPT_MSG • REJECT_MSG • SYNCHRONISED_MSG <p>The HashMap that is returned contains a list of error messages (if any). A successful operation returns an empty map.</p>
GTINTree MessageItemHandler::getHierarchyTree()	<pre>msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); out.writeln("Root Tree is " + rootTree);</pre>	<p>This operation can be used to retrieve the CIN represented by the MessageItemHandler as a Tree [GTINtree].</p> <p>Note: One CIN is equivalent to a single MessageItemHandler</p>
Integer GTINTree :: getNumberOfChildren()	<pre>msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree();</pre>	<p>This operation returns number of child GTIN tree(s) for the root item in the GTINTree in context. This returns only the immediate children of the</p>

Messaging		
Script Operation Name	Usage	Description
	<pre> numberOfChildren = rootTree.getNumberOfChildren(); out.writeln("Number of children are " + numberOfChildren); </pre>	<p>root item.</p> <p>Example:</p> <pre> A / \ B C D E </pre> <p>In this case the number of children for A would be returned as 3.</p>
Integer GTINTree :: getNumberOfParent()	<pre> msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); var childTree = rootTree.getChild(0); numberOfParents = childTree.getNumberOfParents(); out.writeln("Number of parents are " + numberOfParents); </pre>	<p>This operation returns number of parent GTIN tree(s) for the root item in the GTINTree in context. It returns the number of immediate parents.</p> <p>Example:</p> <pre> A / \ B C D E </pre> <p>In this case the number of parents for B would be returned as 1.</p>
GTINTree :: getChild(Integer index)	<pre> msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); var childTree = rootTree.getChild(0); out.writeln("child tree at 0 is " + childTree); </pre>	<p>This operation can be used to retrieve the child tree at the specified index.</p>
Integer GTINTree :: getParent(Integer index)	<pre> msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); var childTree = rootTree.getChild(0); </pre>	<p>This operation can be used to retrieve the parent tree at the specified index.</p>

Messaging		
Script Operation Name	Usage	Description
	<pre>var parentTree = childTree.getParent(0); out.writeln("Parent tree at 0 is " + parentTree);</pre>	
Integer GTINTree :: isLeaf()	<pre>msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); var childTree = rootTree.getChild(0); isLeaf = childTree.isLeaf(); out.writeln("Is Child Tree leaf " + isLeaf);</pre>	<p>This operation is used to check whether the root item represented by the GTINTree is a leaf node. This returns true if the item is a leaf node in the complete hierarchy.</p> <p>Example:</p> <pre> A / \ B C D E</pre> <p>In this case the invoking the isLeaf method on the GTINTree represented for item E, B and C would return true.</p>
ITradeItem GTINTree :: getRootTradeItem()	<pre>msgId = "5"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); rootItem = rootTree.getRootTradeItem(); out.writeln("Root Item GTIN is " + rootItem. getTradeItemAttributeValueDS("Glob al_Attributes_Spec/GlobalTradeItem Number"));</pre>	<p>This operation returns the root trade item object [ITradeItem] of the GTINTree in the current context.</p> <p>Note: For List of API's exposed on the TradeItem (like the getTradeItemAttributeValueDS used above) refer GDS Documentation</p>
HashMap MessageItemHandler :: getAssociatedLinks (String gtin, Boolean fetchChildLinks)	<pre>msgId = "3"; msgItemHandler = new MessageItemHandler(msgId); var rootTree = msgItemHandler.getHierarchyTree(); out.writeln("Root Tree is " +</pre>	<p>The getAssociatedLinks () method can be invoked on the MessageItemHandler object. This returns a HashMap of ITradeItemLink objects associated with the specified GTIN.</p>

Messaging		
Script Operation Name	Usage	Description
	<pre> rootTree); childLinks = msgItemHandler.getAssociatedLinks(rootTree.getRootTradeItem().getTradeItemAttributeValuesDS("Global_Attributes_Spec/GlobalTradeItemNumber"), true); out.writeln("Child Links are " + childLinks); out.writeln("First Child Links is " + childLinksList["0"]); </pre>	<p>If the <i>fetchChildLinks</i> flag is true, the list contains ITradeItemLink objects where the specified GTIN is the parent item in the link. If the <i>fetchChildLinks</i> flag is false, the list contains ITradeItemLink objects where the specified GTIN is the child item in the link. [Note: A TradeItemLink Object represents a single link ex: A (parent)---B(child)].</p> <p>Example</p> <pre> A / \ B C D E </pre> <p>In this case when <i>getAssociatedLinks</i> is called for item A and <i>fetchChildLinks</i> is true, the following TradeItemLink object will be returned:</p> <ol style="list-style-type: none"> 1. A-B 2. A-C 3. A-D <p>Note: In case no item links are present, NULL will be returned. [ex: if <i>fetchChildLinks</i> is false for the above case, NULL will be returned]</p>
String TradeItemLink::getParentGTIN()	<pre> link = getTradeItemLink("<parent GTIN>", "<child GTIN>"); parent = link.getParentGTIN(); out.writeln ("Parent GTIN "+parent); </pre>	This operation returns the parent GTIN of the given Trade Item Link
String TradeItemLink::	<pre> link = getTradeItemLink("<parent </pre>	This operation returns the

Messaging		
Script Operation Name	Usage	Description
getChildGTIN()	<pre>GTIN>", "<child GTIN>"); parent = link.getChildGTIN(); out.writeln ("Parent GTIN "+parent);</pre>	child GTIN of the given Trade Item Link

Script operations related to Spec Map

Script Operation Name	Prototype	Description
getMappings	getMappings()	Returns a hashmap of destination paths to their corresponding source paths in a specmap.

Using Script Operations

This section provides the sample implementations for trade items, trading partners, and trade item links.

Trade Items

The sample scripts for the following are provided:

- Create a New Trade Item, Create Transaction and Notification
- Global Enrichment of a Trade Item
- Getting a Trade Item
- Global and Local Enrichment of Trade Item
- Setting Attributes to a Trade Item
- Getting Attribute Values of a Trade Item
- Search Trade Items
- Get Modified Trade Items Between Dates
- Validating Trade Item
- Get Trade Items
- Get All TradeItem Attributes



Create a New Trade Item, Create Transaction, and Notification

The script operations used are new TradeItem, saveTradeItem, createTransaction, and createNotification.

```
tradeItem = new
TradeItemDS("12345678908765", "U2", "FMCG", "CASE", "10000056", "768576
35353536", "Test", "000000001.000000002.000000012");

    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/P
ack", "8");
tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GTINName
/language", "en_US");
tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GTINName
/text", "Test");
    out.writeln(tradeItem);
    out.writeln(tradeItem.saveTradeItemDS());

    var txnCreated =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mast
er");
    out.writeln("Transaction Created ? : "+txnCreated);

    var notfCreated =
createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mas
ter");
    out.writeln("Notification sent ? : "+notfCreated);
```

Global Enrichment of a Trade Item

The script operations used are getTradeItem, setTradeItemAttribute, saveTradeItem, createTransaction, and createNotification.

```
var tradeItem =
getTradeItemDS("12345678908765", "XX", "XX", "76857635353536");
tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/tradeIte
mInformation/tradeItemDescriptionInformation/brandName/en_US", "Coc
a Cola");
tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/Pack", "8
");
tradeItem.setTradeItemAttributeDS("FMCG_Global_Attributes_Spec/trade
ItemIdentification/TradeItemIdentificationListType", "CA");
tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemS
tatus", "Global Enrichment Completed");

var itemSaved = tradeItem.saveTradeItemDS();
out.writeln("TradeItem [44444444444442] Saved ? : "+itemSaved);

var txnCreated =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "ma
ster");
out.writeln("Transaction Created ? : "+txnCreated);

var notfCreated =
createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "m
aster");
out.writeln("Notification sent ? : "+notfCreated);
```

Getting a Trade Item

```
var tradeItem =
getTradeItemDS("12345678908765", "XX", "XX", "76857635353536");
```



```

out.writeln(tradeItem);

//Global and Local Enrichment of Trade Item
tradeItem = new
TradeItemDS("46456456","U2","FMCG","CASE","10000056","936534534534
51","Test","000000001.000000002.000000012");
if(tradeItem != null)
{
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/t
radeItemInformation/tradeItemDescriptionInformation/brandName/en_U
S","Coca Cola");
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/P
ack","8");
    tradeItem.setTradeItemAttributeDS("FMCG_Global_Attributes_S
pec/tradeItemIdentification/TradeItemIdentificationListType","CA")
;
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/G
DSItemStatus","Draft With Variant");
    tradeItem.setTradeItemAttributeDS("Global_Local_Attributes_
Spec/TargetMarket","US");
    tradeItem.setTradeItemAttributeDS("Global_Local_Attributes_
Spec/InformationProvider","7981315112271");
    tradeItem.setTradeItemAttributeDS("Global_Local_Attributes_
Spec/InformationProviderName","TEST_GLN");
    tradeItem.setTradeItemAttributeDS("Global_Local_Attributes_
Spec/IsPrivate","true");
    tradeItem.setTradeItemAttributeDS("Global_Local_Attributes_
Spec/GDSItemStatus","Approved");

    var itemSaved = tradeItem.saveTradeItemDS();
    out.writeln("TradeItem [45645666666660] Saved ? :
"+itemSaved);

    var txnCreated =
createTransaction(tradeItem,"NEW_ITEM_APPROVED","master");
    out.writeln("Transaction Created ? : "+txnCreated);

    var notfCreated =
createNotification(tradeItem,"NEW_ITEM_APPROVED","master");
    out.writeln("Notification sent ? : "+notfCreated);
}

```

Setting Attributes to a Trade Item

```

//The script operation used is settradeitemattributesDS().
var tradeItem =
getTradeItemDS("46456456","US","7981315112271","93653453453451");
if(tradeItem != null)
{
    var hmAttrPathsToValues = [];
    hmAttrPathsToValues["Global_Attributes_Spec/GTINName/en_US"
] = "Manju23";
    hmAttrPathsToValues["Global_Local_Attributes_Spec/PartnerSp
ecificAttributes/Attr1"] = "Test Description5";
    tradeItem.setTradeItemAttributesDS(hmAttrPathsToValues);
    var itemSaved = tradeItem.saveTradeItemDS();
    out.writeln("TradeItem [46456456] Saved ? : "+itemSaved);
}

```



Getting Attribute Values of a Trade Item

```
var tradeItem =
getTradeItemDS("46456456","US","7981315112271","93653453453451");
if(tradeItem != null)
{
    var gtinName =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Attributes_Spec/GTINName/text"),"");
    var ip =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/InformationProvider"),"");

    out.writeln("GTINName is "+gtinName);
    out.writeln("Attr2 is "+attr1);
    out.writeln("ipName is "+ipName);
}
```

Search Trade Items

```
var hmTradeItems = [];
var hmSearchKeysToValues = [];

hmSearchKeysToValues["EAN_UCC_CODE"] = "46456456";
hmSearchKeysToValues["GTIN"] = "93653453453451";
hmSearchKeysToValues["UDEX_CODE"] =
"000000001.000000002.000000012*";

hmTradeItems = searchTradeItemsDS(hmSearchKeysToValues);

if(hmTradeItems != null)
{
    out.writeln("Returned Trade Items are \n");
    for(i = 0; i < hmTradeItems.size(); i++)
    {
        out.writeln(hmTradeItems[i]);
    }
}
else
{
    out.writeln("No Trade Items found in Search");
}
```



Get Modified Trade Items between Dates

```
var txnType = "Global Attribute Enrichment";
var startDate = new Date("dd-MMM-yyyy", "07-OCT-2004");
var endDate = new Date("dd-MMM-yyyy", "09-OCT-2004");

var hmTradeItems = [];
hmTradeItems = getModifiedTradeItems(txnType,startDate,endDate);

if(hmTradeItems != null)
{
    out.writeln("Returned Trade Items are \n");
    for(i = 0; i < hmTradeItems.size(); i++)
    {
        out.writeln(hmTradeItems[i]);
    }
}
else
{
    out.writeln("No Trade Items found in Search");
}
```

Validating Trade Item

```
var tradeItem =
getTradeItemDS("46456456", "US", "7981315112271", "93653453453451");
hmErrors = tradeItem.validateTradeItem();

out.writeln(hmErrors);
```

Get Trade Items

```
var tradeItem =
getTradeItemDS("46456456", "US", "7981315112271", "93653453453451");
var items = tradeItem.getItems();

out.writeln(items);
```

Get All TradeItem Attributes

```
var tradeItem =
getTradeItemDS("46456456", "US", "7981315112271", "93653453453451");

var hmAttrPathsToValues = tradeItem.getAllTradeItemAttributes();

forEachHmElement(hmAttrPathsToValues, attrPath, attrValue)
{
    out.writeln(attrPath + " : "+attrValue);
}
```

}

Trading Partners

The scripts are provided for the following:

- Create a New Trading Partner
- Update an Existing Trading Partner
- Get Trading Partner Attribute Value
- Search Trading Partner

Create a New Trading Partner

```
var tradingPartner = new
TradingPartner("6701115112124", "TEST_GLN", "US", "Supplier");
tradingPartner.setPartnerAttribute("GLN_Spec/Status", "Approved");
var tpSaved = tradingPartner.saveTradingPartner();

out.writeln("TradingPartner [6701115112124] Saved ? : "+tpSaved);

var txnCreated =
createTransaction(tradingPartner, "NEW_TRADING_PARTNER_APPROVED", "m
aster");
out.writeln("Transaction Created ? : "+txnCreated);

var notfCreated =
createNotification(tradingPartner, "NEW_TRADING_PARTNER_APPROVED", "
master");
out.writeln("Notification sent ? : "+notfCreated);
```

Update an Existing Trading Partner

```
var tradingPartner = getTradingPartner("6701115112124");
var hmAttrPathsToValues = [];
hmAttrPathsToValues["GLN_Spec/Trading Partner State"] = "CA";
tradingPartner.setPartnerAttributes(hmAttrPathsToValues);

var tpSaved = tradingPartner.saveTradingPartner();
out.writeln("TradingPartner [6701115112124] Saved ? : "+tpSaved);

var txnCreated =
createTransaction(tradingPartner, "MODIFY_TRADING_PARTNER_APPROVED"
, "master");
out.writeln("Transaction Created ? : "+txnCreated);

var notfCreated =
createNotification(tradingPartner, "MODIFY_TRADING_PARTNER_APPROVED
", "master");
out.writeln("Notification sent ? : "+notfCreated);
```

Get Trading Partner Attribute Value

```
var tradingPartner = getTradingPartner("6701115112124");
```

```

var tpLanguage =
tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner
Language");
var tpID = tradingPartner.getPartnerAttributeValue("GLN_Spec/GLN
Identifier");
var tpState =
tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner
State");
out.writeln("Trading Partner language is "+tpLanguage);
out.writeln("Trading Partner ID is "+tpID);
out.writeln("Trading Partner State is "+tpState);

```

Search Trading Partners

```

var hmSearchKeysToValues = [];
hmSearchKeysToValues["PARTNER_NAME"] = "TEST_GLN";
var hmTradingPartners =
searchTradingPartners(hmSearchKeysToValues);

if(hmTradingPartners != null)
{
    out.writeln("Returned Trading Partners are \n");
    for(i = 0; i < hmTradingPartners.size(); i++)
    {
        out.writeln(hmTradingPartners[i]);
    }
}
else
{
    out.writeln("No hmTradingPartners found in Search");
}

```

Trade Item Links

The scripts are provided for the following:

- Create a New Trade Item Link
- Update Trade Item Links
- Get Trade Item Links Attributes
- Get Child Links
- Get Parent Links

Create a New Trade Item Link

```

var tradeItem = new
TradeItemDS("45645666666677", "CHILD", "EACH", "10000056", "000000001.
000000002.000000012");

var itemSaved = tradeItem.saveTradeItemDS();
out.writeln("TradeItem [45645666666677] Saved ? : "+itemSaved);

var txnCreated =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_PENDING", "mast
er");
out.writeln("Transaction Created ? : "+txnCreated);

```



```
var notifCreated =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_PENDING","mas
ter");
out.writeln("Notification sent ? : "+notifCreated);

var tradeItemLink = new
TradeItemLink("4564566666660","45645666666677",1);

var linkSaved = tradeItemLink.saveTradeItemLink();
out.writeln("4564566666660 < == > 45645666666677 [LINK] Saved ?
:"+linkSaved);
var bCreateTxn =
createTransaction(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING
","master");
out.writeln("Transaction Created ? : "+bCreateTxn);
var bCreateNotf =
createNotification(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDIN
G","master");
out.writeln("Notification sent ? : "+bCreateNotf);
```

Update Trade Item Links

```
var tradeItemLink =
getTradeItemLink("4564566666660","45645666666677");
tradeItemLink.setLinkAttribute("Status","Approved");
var hmAttrsToValues = [];
hmAttrsToValues["Quantity"] = "2";
tradeItemLink.setLinkAttributes(hmAttrsToValues);
var linkSaved = tradeItemLink.saveTradeItemLink();
out.writeln("4564566666660 < == > 45645666666677 [LINK] Saved ?
:"+linkSaved);
var bCreateTxn =
createTransaction(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING
","master");
out.writeln("Transaction Created ? : "+bCreateTxn);
var bCreateNotf =
createNotification(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDIN
G","master");
out.writeln("Notification sent ? : "+bCreateNotf);
```

Get Trade Item Links Attributes

```
var tradeItemLink =
getTradeItemLink("4564566666660","45645666666677");

var qty = tradeItemLink.getLinkAttributeValue("Quantity");
var status = tradeItemLink.getLinkAttributeValue("Status");
var childGtin =
tradeItemLink.getLinkAttributeValue("GlobalTradeItemNumber");

out.writeln("Quantity is "+qty);
out.writeln("Status is "+status);
out.writeln("ChildGtin is "+childGtin);
```

Get Child Links

```
var tradeItem =
getTradeItemDS("4564566666660","US","7981315112271");
```



```
var hmChildLinks = [];  
hmChildLinks = tradeItem.getTradeItemChildLinks();  
  
if(hmChildLinks != null)  
{  
    out.writeln("Returned Child Trade Item Links are \n");  
    for(i = 0; i < hmChildLinks.size(); i++)  
    {  
        out.writeln(hmChildLinks[i]);  
    }  
}  
else  
{  
    out.writeln("No Child TradeItem Links found in Search");  
}
```

Get Parent Links

```
var tradeItem = getTradeItemDS("45645666666677","XX","XX");  
  
var hmParentLinks = [];  
hmParentLinks = tradeItem.getTradeItemParentLinks();  
  
if(hmParentLinks != null)  
{  
    out.writeln("Returned Parent Trade Item Links are \n");  
    for(i = 0; i < hmParentLinks.size(); i++)  
    {  
        out.writeln(hmParentLinks[i]);  
    }  
}  
else  
{  
    out.writeln("No Parent TradeItem Links found in Search");  
}
```

Spec Maps

```
specMap = getSpecMapByName("GDS XML GLN Attributes Spec Map");  
var hmDstPathsToSrcPaths = specMap.getMappings();  
  
out.writeln(hmDstPathsToSrcPaths);
```



Notices

IBM Global Data Synchronization for WebSphere Product Center includes the following software that was licensed by IBM from the Apache Software Foundation under the terms and conditions of the Apache 2.0 license:

- Apache Struts v1.1
- Apache Commons-beanutils v1.6
- Apache Commons-digester v1.4
- Apache Xerces-j v2.4.0
- Apache XMLBeans v1.0.3
- Apache Commons-JXPath v1.2
- XML4J 4.3.3

This software is licensed to you under the terms and conditions of the International Program License Agreement, subject to its Excluded Components provisions. IBM is required to provide the following notice to you in connection with this software:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.



"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of



this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.



You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only



on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

IBM Global Data Synchronization for WebSphere Product Center also includes the following software, portions of which were licensed from the World Wide Web Consortium (W3C):

- Apache XMLBeans v1.0.3
- Apache Commons-JXPath v1.2

This software is licensed to you under the terms and conditions of the International Program License Agreement, subject to its Excluded Components provisions. IBM is required to provide the following notice to you in connection with portions of this software:

W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>



This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.



Joseph Reagle <site-policy@w3.org>

Last revised \$Id: copyright-software-20021231.html,v 1.11 2004/07/06 16:02:49
slesch Exp \$

IBM Global Data Synchronization for WebSphere Product Center also includes the following software:

- XML4J 4.3.3

This software is licensed to you under the terms and conditions of the International Program License Agreement, subject to its Excluded Components provisions. IBM is required to provide the following notice to you in connection with portions of this software:

ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE Copyright (c) 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

