



IBM Global Data Synchronization for WebSphere Product Center  
Version 1.1.1

# Installation, Configuration and Administration Guide

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

23 February 2005

This edition of this document applies to IBM Global Data Synchronization for Websphere Product Center, version 1.1.1, and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporations 2004, 2005. All rights reserved.

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

# Table of Contents

<b>ABOUT THIS DOCUMENT.....</b>	<b>1</b>
AUDIENCE .....	1
RELATED DOCUMENTATION .....	1
TYPOGRAPHIC CONVENTIONS .....	1
<b>CH 1    OVERVIEW.....</b>	<b>3</b>
<b>CH 2    INSTALLATION.....</b>	<b>4</b>
SYSTEM REQUIREMENTS .....	4
<i>AIX requirements</i> .....	4
<i>WebSphere Application Server/DB2 on AIX</i> .....	5
<i>WebSphere Application Server/Oracle on AIX</i> .....	5
<i>Linux requirements</i> .....	6
<i>WebSphere Application Server/DB2 on Linux</i> .....	6
<i>WebSphere Application Server/Oracle on Linux</i> .....	7
<i>WebLogic/Oracle on Linux</i> .....	7
<i>HP-UX requirements</i> .....	8
<i>WebLogic/Oracle on HP-UX</i> .....	8
<i>Solaris Requirements</i> .....	9
<i>Client requirements</i> .....	9
INSTALLATION PRE-REQUISITES .....	9
INSTALLATION PROCEDURE .....	11
<i>Using Command Line</i> .....	11
<i>Using Installer</i> .....	12
POST INSTALLATION TASKS .....	13
<i>Configuration changes for WebSphere Product Center 5.1.0.3</i> .....	13
<i>Post Installation Tasks for GDS</i> .....	14
<i>Locale/Language Specific Configurations</i> .....	19
<i>Configure DataPool Catalog for supply side</i> .....	19
<i>Configure DataPool Catalog for demand side</i> .....	20
IBM GLOBAL DATA SYNCHRONIZATION FOR WEBSHERE PRODUCT CENTER UNINSTALLATION.....	21
<i>Using Command Line</i> .....	21
<i>Using Installer</i> .....	21
<b>CH 3    CONFIGURATION .....</b>	<b>23</b>
MODIFY OR MOVE THE NAMES OF ATTRIBUTES .....	23
<i>Modify Global Attributes</i> .....	23
<i>Modify Global-Local Attributes</i> .....	24
MODIFY OR MOVE THE NAMES OF ATTRIBUTE GROUPS.....	24
CONFIGURE TARGET MARKETS .....	25
CONFIGURE LANGUAGE SPECIFIC ATTRIBUTES.....	25
CONFIGURE TRADE ITEM STATUS .....	26
CONFIGURE ENRICHMENT COMPLETION CRITERIA .....	26
CONFIGURE APPROVAL PROCESS .....	27
CONFIGURE CLASSIFICATION SCHEME .....	27
CONFIGURE USERS AND ROLES .....	27
CONFIGURE WORKFLOWS .....	28
CONFIGURE INFORMATION PROVIDER.....	28
CONFIGURE TRADE ITEM DATA COMPLIANCE CHECK .....	29
MAPPING BETWEEN INTERNAL CATEGORY AND THE UDEX CATEGORY.....	29
CONFIGURE INTEROPERABILITY ATTRIBUTES- (ONLY SUPPLY SIDE) .....	30
<b>CH 4    ADMINISTRATION.....</b>	<b>32</b>

SECURITY .....	32
<i>Functional Security</i> .....	32
<i>Data Security</i> .....	32
SECURITY STEPS.....	32
<i>Create a Role</i> .....	33
<i>Create an Access Control Group (ACG)</i> .....	33
<i>Create Selections</i> .....	34
<i>Provide Data Access Privileges</i> .....	37
<i>Create User</i> .....	37
<b>CH 5    IMPORT/EXPORT – SUPPLY SIDE .....</b>	<b>38</b>
FILE FORMATS FOR IMPORT/EXPORT.....	38
<i>Implementation Approach for CSV</i> .....	38
<i>Implementation Approach for XML</i> .....	39
HANDLING VALIDATIONS.....	40
SCENARIOS AND SAMPLE SCRIPTS.....	41
<i>For Import</i> .....	41
<i>For Export</i> .....	71
SCRIPT TABLE .....	86
USING SCRIPT OPERATIONS.....	90
<i>Trade Items</i> .....	90
<i>Trading Partners</i> .....	94
<i>Trade Item Links</i> .....	95
<i>Spec Maps</i> .....	97
<b>CH 6    IMPORT/EXPORT –DEMAND SIDE .....</b>	<b>98</b>
FILE FORMATS FOR IMPORT/EXPORT.....	98
<i>Implementation Approach for CSV</i> .....	98
<i>Implementation Approach for XML</i> .....	99
SCENARIOS AND SAMPLE SCRIPTS.....	100
<i>For Import</i> .....	100
<i>For Export</i> .....	129
SCRIPT TABLE .....	149
USING SCRIPT OPERATIONS.....	154
<i>Trade Items</i> .....	154
<i>Trading Partners</i> .....	158
<i>Trade Item Links</i> .....	159
<i>Spec Maps</i> .....	161
<b>APPENDIX A – PRECEDED ROLES, ACCESS CONTROL GROUPS, AND SELECTIONS – SUPPLY SIDE</b>	<b>163</b>
ROLES .....	163
WORKFLOW ROLES .....	179
ACCESS CONTROL GROUPS .....	180
SELECTIONS .....	182
<b>APPENDIX B – PRE-SEEDED ROLES, ACCESS CONTROL GROUPS, AND SELECTIONS –DEMAND SIDE</b>	<b>183</b>
ROLES .....	183
WORKFLOW ROLES .....	197
ACCESS CONTROL GROUPS .....	199
SELECTIONS .....	201

---

# About This Document

## ***Audience***

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

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
- 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 Overview

Global Data Synchronization is the process by which trading partners exchange basic information about products on an ongoing basis. The information can be on price, party, and other relationship-specific details. This process ensures that all stakeholders for a product are kept in “sync” about the latest information on the product.

Global Data Synchronization enables you to:

- Reduce costs and increase competitiveness through the efficient and accurate transfer and on-going synchronization of information.
- Lay the foundation for future supply chain collaboration initiatives which require that trading partners are in “sync” on the basic product, price, party, and relationship specific information.

Note: This document explains procedures for installing and configuring IBM Global Data Synchronization for WebSphere Product Center solution for both Demand and Supply component users. Follow instructions that are applicable to you depending on whether you are installing a demand side component or a supply side component.

---

## Ch 2 Installation

This chapter provides complete instructions for installing the solution along with a list of pre-requisites and the necessary hardware and software components.

Topics:

- [System Requirements](#)
- [Installation Pre-requisites](#)
- [Supported Platforms](#)
- [Installation](#)
- [Post Installation Tasks](#)
- [Uninstallation](#)

### **System Requirements**

This chapter details the minimum hardware and software required for installing the IBM Global Data Synchronization for WebSphere Product Center solution.

Note: The requirements listed in this section are the minimum hardware and software recommendations. Prior to installation, a requirement analysis is recommended by WebSphere Professional Services to plan hardware considerations based on the choice of software used and the projected load volume of the servers (that is, the number of SKU's and transaction rates.)

### **AIX requirements**

This section outlines the hardware and software requirements for the following platform:

Operating System	Application Server	Database Server
AIX Version 5L v5.2	IBM WebSphere Application Server Network Deployment Version 5.0.2.5	IBM DB2 II Version 8.1 Advanced Edition, Fix Pack 5 OR Oracle 9.2.0.5 Enterprise Edition



## WebSphere Application Server/DB2 on AIX

Application Server	
Software	WebSphere Application Server Network Deployment 5.0.2.5 AIX 5L 5.2 (32 & 64 bit) IBM HTTP Server 2.0 IBM JDK bundled with WebSphere Application Server Network Deployment 5.0.2.5
Hardware	Server Type – IBM p630 Model 6C4 Processor – 4 x 64-bit POWER4+ 1.45 Memory – 8 GB Disk Sub-System – 144 GB SCSI
Database Server	
Software	DB2 II 8.1 Advanced Edition Fixpack 5 AIX 5L 5.2
Hardware	Server Type – IBM p630 Model 6C4, PE2650 Processor – 4 x 64-bit POWER4+ 1.45 GHz Memory – 8 GB Disk Sub-System – 420 GB SCSI
Additional Software	
<p>This primary stack is also bundled with the following:</p> <p>WebSphere MQ Series V5.3 (1)</p> <p>WebSphere Business Integration Adapter for WebSphere MQ V2.5.0 (1, 2)</p> <p>WebSphere Business Integration Adapter Framework V2.4.0 (1, 2)</p> <p>(1) A limited use license is included with WebSphere Product Center</p> <p>(2) Required when integrating with a process hub</p>	

## WebSphere Application Server/Oracle on AIX

Application Server	
Software	WebSphere Application Server Network Deployment 5.0.2.5 AIX 5L 5.2 IBM HTTP Server 2.0 IBM JDK bundled with WebSphere Application Server Network Deployment 5.0.2.5
Hardware	Server Type – IBM p630 Model 6C4

	Processor – 4 x 64-bit POWER4+ 1.45 GHz Memory – 8 GB Disk Sub-System – 144 GB SCSI
<b>Database Server</b>	
Software	Oracle 9.2.0.5 Enterprise Edition AIX 5L 5.2
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz Hardware Processor: Dual Intel, XEON 2.8 GHz Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver

## Linux requirements

Operating System	Application Server	Database Server
Red Hat Enterprise Linux ES 2.1	WebSphere Application Server 5.0.2.5	IBM DB2 II Version 8.1 Advanced Edition, Fix Pack 5 OR Oracle 9.2.0.5
Red Hat Enterprise Linux ES 2.1	WebLogic Server 7.0	Oracle 9.2.0.5 Enterprise Edition

## WebSphere Application Server/DB2 on Linux

<b>Application Server</b>	
Software	WebSphere Application Server Network Deployment 5.0.2.5 IBM HTTP Server 2.0 Red Hat Enterprise Linux ES 2.1 IBM JDK bundled with WebSphere Application Server Network Deployment
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz

	Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver
<b>Database Server</b>	
Software	DB2 II 8.1 Advanced Edition Fixpack Red Hat Enterprise Linux ES 2.1
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver

### WebSphere Application Server/Oracle on Linux

<b>Application Server</b>	
Software	WebSphere Application Server Network Deployment 5.0.2.5 IBM HTTP Server 2.0 Red Hat Enterprise Linux ES 2.1 IBM JDK bundled with WebSphere Application Server Network Deployment
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver
<b>Database Server</b>	
Software	Oracle 9.2.0.5 Enterprise Edition Red Hat Enterprise Linux ES 2.1
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver

### WebLogic/Oracle on Linux

<b>Application Server</b>	
Software	WebLogic Server 7.0 Apache HTTP Server 2.0

	Red Hat Enterprise Linux ES 2.1 WebLogic JDK bundled with WebLogic Server 7.0
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver
<b>Database Server</b>	
Software	Oracle 9.2.0.5 Enterprise Edition Red Hat Enterprise Linux ES 2.1
Hardware	Server Type: Intel compatible Processor: Dual Intel, XEON 2.8 GHz Memory: 6 GB Disk Sub-System: 365 GB 10K RPM SCSI Driver

## HP-UX requirements

Operating System	Application Server	Database Server
HP UX 11i	WebLogic Server 7.0	Oracle 9.2.0.5 Enterprise Edition

## WebLogic/Oracle on HP-UX

<b>Application Server</b>	
Software	WebLogic Server 7.0 Apache HTTP Server 2.0 HP/UX 11i WebLogic JDK bundled with WebLogic Server 7.0
Hardware	Server Type: HP RP 4440 Processor: 2 CPUs Processor: 2 CPUs Memory: 4 GB Disk Sub-System: HP MSA 1000
<b>Database Server</b>	
Software	Oracle 9.2.0.5 Enterprise Edition

	HP/UX 11i
Hardware	Server Type: HP RP 4440 Processor: 2 CPUs Memory: 4 GB Disk Sub-System: HP MSA 1000

## Solaris Requirements

Operating System	Application Server	Database Server
Solaris 9	WebSphere Application Server 5.0.2.5	Oracle 9.2.0.5

## Client requirements

Operating environment	
Software	Microsoft Windows 2000 Professional Service Pack 3 OR Microsoft Windows XP Professional Service Pack 1 AND Browser: Microsoft Internet Explorer Version 6.0 Service Pack 1
Hardware	Processor: Intel-based system with a Pentium 3 processor and a minimum processor speed of 750 MHz Memory: 256 MB

Note: If your IE browser displays any other tool bar other than a IE tool bar( for example, the Yahoo toolbar), then error messages such as “Page not found” may be displayed when working with the IBM Global Data Synchronization for WebSphere Product Center solution. Therefore, ensure that you do not have any other toolbars other than IE toolbars. Also ensure that no other utilities are installed on your IE browser.

## Installation Pre-requisites

This section provides the pre-requisites necessary before installing the IBM Global Data Synchronization for WebSphere Product Center solution.

Prior to installing IBM Global Data Synchronization for WebSphere Product Center application, ensure that the WebSphere Product Center version 5.1.0.3 is installed on your system. Download WPC binaries (currently for 5.1.0.3 with all patches) and put them in the

source directory for the user. For information about installing WebSphere Product Center version 5.1.0.3, refer *WebSphere Product Center 5.1 Installation Guide*.

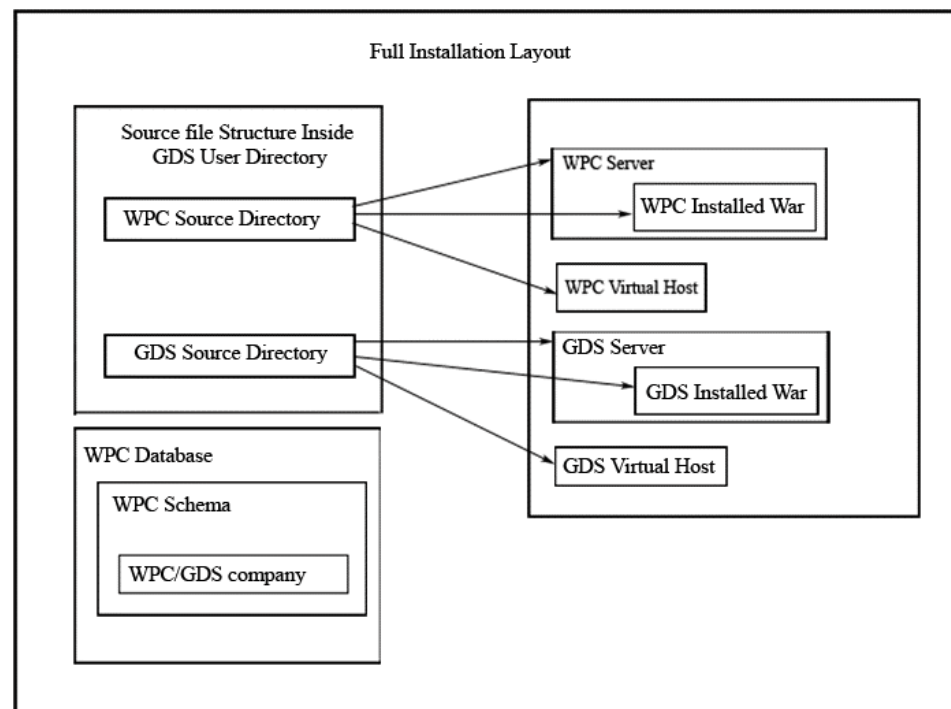
The pre-requisites are:

1. WPC 5.1.0.3 Patch must be applied with the necessary schema updates as required.
2. WPC 5.1.0.3 should be installed and configured. WPC and GDS both require a server, a virtual host, and an installed war file. Neither are these components are shared nor can the WPC and GDS directory be shared.
3. Company should be created other than IBM. For example, Walmart. GDS and WPC will be using the same schema and company name.

The following diagram gives an overview of the installation layout. There are two source file directories; one for WebSphere Product Center files, and the other for IBM Global Data synchronization for WebSphere Product Center files.

There are two separate application servers created for each of the applications on which the respective war files are installed. Similarly there are two virtual hosts created.

The database is shared by both the applications. Both the applications use the same company.



*GDS – WPC intallation layout*

Note: WebSphere Business Integration Connect(WBIC) and IBM Global Data Synchronization for WebSphere Product Center should each be deployed in operating environments consistent with the technology platforms supported by the individual products. There is no technology platform dependency between these two products.

## **Installation procedure**

This chapter provides the installation procedure to install the IBM Global Data Synchronization for WebSphere Product Center solution. Before installing the IBM Global Data Synchronization for WebSphere Product Center, ensure that WebSphere Product Center is not running.

The installation process is described primarily for AIX operating system, DB2 database and WebSphere Application Server. The instructions hold good for all installation on currently supported platforms, database and application servers. Depending on which combination of Operating system, Database and Application Server, specific changes are mentioned as a note.

The IBM Global Data Synchronization for WebSphere Product Center solution can be installed in two ways:

- Using Command Line
- Using Installer

### **Using Command Line**

WebSphere Product Center can be installed from the command line. To install silently:

```
./setupAIX.bin -silent -P installLocation=<value> <PRODUCT_SET_FLAG>
```

where <PRODUCT\_SET\_FLAG> is as follows:

for AIX/WAS/DB2:	-W adapterFeatures_aix.selectedFeature=0
for AIX/WAS/Oracle:	-W adapterFeatures_aix.selectedFeature=1
for Linux/WAS/DB2:	-W adapterFeatures_linux.selectedFeature=0
for Linux/WAS/Oracle:	-W adapterFeatures_linux.selectedFeature=1
for Linux/Weblogic/Oracle:	-W adapterFeatures_linux.selectedFeature=2
for Solaris/Weblogic/Oracle:	-W adapterFeatures_solaris.selectedFeature=0
for HP/Weblogic/Oracle:	-W adapterFeatures_hp.selectedFeature=0

The following example will install the first AIX code distribution (WAS/DB2) of GDS in directory /opt/IBM/WebSphereProductCenter:

```
./setupAIX.bin -silent -W adapterFeatures_aix.selectedFeature=0 -P installLocation  
=/opt/IBM/WebSphereProductCenter
```

Optionally silent install can be performed using an options file:

```
./setupAIX.bin -options <options file name> -silent
```

For the example stated above, the options file will need to contain the following lines:

```
-W adapterFeatures_aix.selectedFeature=0  
-P installLocation=<value>
```

The silent install will not prompt the user to accept the license agreement. In addition, it does not require a command-line flag that indicates acceptance of the license agreement before the product installs. This is the expected behavior of the IBM component software that performs the license agreement acceptance, and we are required to embed it within the installer.

Note: If the installation is being done on a Linux box, then use `setupLinux.bin` as the installer file.

If the installation is being done on HP UX, then use `setupHP.bin` as the installer file.

Note: Sometimes the long name of the default directory path might cause some discrepancies. Hence to be on the safer side rename the `$TOP` directory to something shorter like `"gds_demand"` or `"gds_supply"`.

## Using Installer

The Installer is used to unzip the files to the required directory.

To install the IBM Global Data Synchronization for WebSphere Product Center solution using the installer:

1. Untar the `.tar` file and run the `setupAix.bin` executable available in `GDS_Demand_AIX` directory. The *Splash* window is displayed for a few seconds and then the *Welcome* window is displayed. This provides a basic introduction to the installation process. The product name, version, vendor name, and the Website information are also provided.

Note: If the installation is being done on a Linux box, then use `setupLinux.bin` as the installer file.

If the installation is being done on HP UX box, then use `setupHP.bin` as the installer file.

2. At the "Welcome" screen click **Next**.
3. At the IBM software license acceptance screen, click "I accept the terms in the license agreement" and then click **Next**.
4. Select the application server and database combination from the available list of combinations and click **Next**.
5. At the product directory screen, perform one of the following tasks:
  - a. Type the full path of the directory into which you want to install WebSphere Product Center in the Directory Name field and click **Next**.
  - b. Click Browse to select a directory and click **Next**.
  - c. Accept the default directory and click **Next**.
6. The summary screen lists the features that will be installed, the specified product directory, and the amount of disk space required.



7. Click **Next**.
8. After the installer finishes successfully, click **Finish**. This completes the installation process.

### Log Files

The installer log file is `installLogGDS.txt`. The installed and uninstalled components are logged in this file. The location of the file is <install location> directory.

---

## Post Installation Tasks

This section provides the post installation tasks for WebSphere Product Center 5.1.0.3 and IBM Global Data Synchronization for WebSphere Product Center.

### Configuration changes for WebSphere Product Center 5.1.0.3

The following configuration changes should be made in WebSphere Product Center 5.1.0.3 instance after installing IBM Global Data Synchronization for WebSphere Product Center:

1. Copy `gds.properties` located in `$TOP/etc/default/` directory in the IBM Global Data Synchronization for WebSphere Product Center instance to `WPC_$TOP/etc/default/` directory in the WebSphere Product Center 5.1.0.3 instance.

Or

Give a link to `gds.properties` in IBM Global Data Synchronization for WebSphere Product Center instance.

Example on AIX:

```
ln -s /usr/qa/gds/gds_demand/etc/default/gds.properties gds.properties
```

2. Copy `gdsTradeItemFunctions.properties` located in `$TOP/etc/default/` directory in the IBM Global Data Synchronization for WebSphere Product Center instance to `WPC_$TOP/etc/default/` directory in the WebSphere Product Center 5.1 instance.

Or

Give a link to `gdsTradeItemFunctions.properties` in IBM Global Data Synchronization for WebSphere Product Center instance.

Example on AIX:

```
ln -s /usr/qa/gds/gds_demand/etc/default/gdsTradeItemFunctions.properties gdsTradeItemFunctions.properties
```

3. Update the path of the following `.jar` in the `ccd_rc` file of WebSphere Product Center 5.1.0.3 located in `$TOP/setup` directory.files and add the `.jar` files t the `$TOP` directory

For supply side intallation update the path of the following `.jar` files

- `cgd_svr.jar`

For demand side intallation update the path of the following `.jar` files

- cgd\_svr.jar
- com.ibm.mqbind.jar
- connector.jar
- fscontext.jar
- jms.jar
- jndi.jar
- providerutil.jar

Example on AIX:

```
AddJar /usr/qa/gds/gds_demand/jars/cgd_svr.jar
```

```
AddJar /usr/qa/gds/gds_demand/jars/com.ibm.mqbind.jar
```

4. Also, manually update this path in CLASSPATH of the appsvr created for WebSphere Product Center 5.1.0.3 from WebSphere Application Server console. Restart WebSphere Product Center 5.1.0.3.
5. Copy .bindings file located in \$TOP/etc/default/ directory in the IBM Global Data synchronization for WebSphere Product Center instance to WPC \$TOP/etc/default/ directory in the WebSphere Product Center 5.1.0.3 instance.

## Post Installation Tasks for GDS

The post installation tasks that are required to be done once IBM Global Data Synchronization for WebSphere Product Center solution is installed are:

1. In the .bashrc file of the IBM Global Data Synchronization for WebSphere Product Center solution, make the following changes:

- a. Point \$TOP to the GDS directory.
- b. Add a parameter export WPC\_INSTALL\_DIR=<Directory where WPC is installed>

Example on AIX:

```
export WPC_INSTALL_DIR=/usr/gds/wpc_5002_07_websphere5_db2_unix
```

- c. Add a parameter WPC\_JAR\_DIR=<Directory where WPC jars are located>

Example on AIX:

```
export WPC_JAR_DIR=/usr/gds/wpc_5002_07_websphere5_db2_unix/jars
```

-Or-

```
export WPC_JAR_DIR=WPC_INSTALL_DIR/jars
```

2. Modify the init\_cgd\_vars.sh file located in \$TOP/setup directory.

If the **installation is on WebSphere Application Server** set the path for the parameters JDK\_HOME, WAS\_HOME, WAS\_NODENAME, WAS\_APPSERVERNAME, and WAS\_VHOST.

Example on AIX:

```
export JDK_HOME=/usr/WebSphere/AppServer/java
export WAS_HOME=/usr/WebSphere/AppServer
export WAS_NODENAME=localhost
export WAS_APPSERVERNAME=gds_reg
export WAS_VHOST=gds_reg
```

Note: Ensure that **the same values are not specified** for WAS\_APPSERVERNAME and WAS\_VHOST as given while installing WebSphere Product Center. Specify different values for WAS\_APPSERVERNAME and WAS\_VHOST for the IBM Global Data Synchronization for WebSphere Product Center instance.

If the **installation is on BEA WebLogic**, set the path for the following

Modify the init\_cgd\_vars.sh file located in \$TOP/setup directory for the parameters WL\_DOMAIN, WL\_HOME, JDK\_HOME, WL\_SERVER, and WL\_USER, WL\_PASSWORD.

Example on AIX:

```
export RMIREGISTRY=($JAVA_HOME/bin)
export JDK_HOME=/root/bea/jrockit70sp5_131_10
export WL_HOME=/root/bea/weblogic700
export CCD_WL_DOMAIN_DIR=/root/bea/user_projects/gdsdmddomain
export CCD_WL_DOMAIN=gdsdmddomain
export CCD_WL_SERVER=gdsserver1
export CCD_WL_PW=gdsuser1
export CCD_WL_USER=gdsuser
```

Note: If the installation is being done for HP UX then in addition to the above variables set the Locale to en\_US.iso88591. For example, export LANG=en\_US.iso88591

Note: Comment the line export APPSVR\_MEMORY\_FLAG='-Xmx1024m -Xms256m -XX:MaxPermSize=128m'.

3. Modify the common.properties file located in \$TOP/etc/default directory for the db connection details, rmi\_port, and app\_svr port.

Note: Ensure that the **same values are specified** for db connection and rmi\_port as given in WebSphere Product Center intallation guide. Since WebSphere Product Center and IBM Global Data Synchronization for WebSphere Product Center use different application servers, specify different value for app\_svr port for the GDS instance.

4. Modify the `gds.properties` file located at `$TOP/etc/default` directory for the `company_code` and `gds_app_type` parameters. The value for `company_code` should be the company name as created in Installation Pre-requisites, Step 3.

Example on AIX,

```
company_code=<Name of company>
```

```
gds_app_type=Demand
```

OR

```
gds_app_type=supply
```

**Note:** The `gds.properties` in `etc/default` directory of the IBM Global Data Synchronization for WebSphere Product Center instance and in the WebSphere Product Center instance should have the same changes. Do the changes in one location and then copy it to the `etc/default` directory of WebSphere Product Center. This is for any future change in `IBM global.properties` also. This is not required if you put a link for `IBM global.properties` in WebSphere Product Center.

5. Run the `test_db.sh` script located in the `$TOP/bin` directory to check for database connection. It should be successful.

For example,

```
bash-2.05b# ./test_db.sh
```

The database connection details are displayed.

6. Configure the Application Server.

To **configure WebSphere Application Server**, run the scripts `create_vhost.sh`, `create_appsvr.sh`, and `install_war.sh` located in the `$TOP/bin/websphere` directory.

**Note:** If you are running Node Manager for WebSphere Application Server, create the `vhost` manually from WAS Admin Console and specify the port on which the IBM Global Data Synchronization for WebSphere Product Center solution will run.

To **configure BEA WebLogic** as the Application Server follow the steps listed in the section

- Set default home directories
- Create a new domain with the `$WL_HOME/common/bin/dmwiz.sh` wizard
- Update the **config.xml** file
- Copy or link the WebLogic domain applications directory

#### Set default home directories

The default home directories are set in step 2 under modifying `init_cgd_vars.sh`.

For the purpose of this document, the WebLogic setup instructions in this section are configured for the following:

- Domain name 'gdsdmdomain'

- Listening on port '7506'
- Domain files contained in the directory '/root/bea/user\_projects/gdsdmddomain'
- Servername set to 'gdsserver1'
- WebLogic username 'gdsuser'
- WebLogic password 'gdsuser1'

### **Create a new WebLogic domain**

To create a new WebLogic domain, use the dmwiz.sh wizard located in the following directory:

`$WL_HOME/common/bin/dmwiz.sh`

- a. **Run** `$WL_HOME/common/bin/dmwiz.sh`
- b. **Choose Domain Type** (3) WLS Domain
- c. **Choose Server Type** (1) Single Server (Standalone Server)
- d. **Choose Domain Name** (1) Modify current selection
- e. **Configure Single Server**
- f. **Modify Server Name** gdsserver1
- g. **Modify Server Listen Port** 7506
- h. **Choose Location to Create Domain > Modify current selection**  
/root/bea/user\_projects
- i. **Enter User Name** gdsuser
- j. **Enter Password** gdsuser1
- k. **Enter Verify Password** gdsuser1
- l. Press Enter to create the domain and follow the prompts to exit the wizard.

At the completion of the dmwiz wizard, the following message is displayed: **Domain Configuration Wizard has successfully created**

**"gdsdmddomain"**

Now that the WebLogic domain has been created in  
"/root/bea/user\_projects/gdsdmddomain ", it can be configured by editing"  
/root/bea/user\_projects/gdsdmddomain /config.xml"

### **Change DefaultWebApp**

Change the DefaultWebApp in the WebServer field from "DefaultWebApp" to  
"demand"

Note: Change it to supply if you are installing the supply component.

### **WebLogic Domain applications directory**

Copy the WebLogic Domain applications directory

Copy (\$TOP/etc/default/weblogic700/config/gdsdmdomain/applications/directory) into the new domain directory.

7. Run the script `./test_java_db.sh` in `$TOP/bin`. After that run the script `create_gds_demand_schema.sh` located in the `$TOP/src/db/schema/create` directory. This loads the data model for IBM Global Data Synchronization for WebSphere Product Center.

For example,

```
create_gds_demand_schema.sh --company_code=<Name of a company> (for demand side installation)
```

```
create_gds_supply_schema.h --company_code=<Name of a company> (for supply side installation)
```

This value should be same as given in Step 4.

**Note:** Before loading the data model, ensure that WebSphere Product Center is not running. Once the data model is loaded, restart WebSphere Product Center.

8. Run the `start_appsvr.sh` script located in the `$TOP/bin/go/start` directory to start the IBM Global Data Synchronization for WebSphere Product Center application. Ensure that WPC instance is running.
9. Access the IBM Global Data Synchronization for WebSphere Product Center application at the `appsvr_port` number as specified in the `common.properties` file. For example, <http://9.182.74.226:9706/Login.do>.

**Note:** If you stop the IBM Global Data Synchronization for WebSphere Product Center solution, it may stop the rmi service of Websphere Product Center. Therefore, ensure that all services are running correctly for WebSphere Product Center.

**Note:**

After loading the datamodel we need to update spec in following way. All following attributes having type as "Number" need to be changed to "string" type . All such attribute should have regular expression `"^[0-9]{1,23}([\.[0-9]{1,15})? $"` attached to it for UI validation.

Attribute	Value
Storage/Handling Temp Min	Global Attributes.
Storage/Handling Temp Max	Global Attributes.
Size[metric]	Global Attributes
Size[imperial]	Global Attributes
Storage Handling Temp[max]	Global Attributes
Storage Handling Temp[min]	Global Attributes
Storage Handling Temp[min]	Global Attributes
Max Admissable Stacking weight	Logistics
Ready to Use volume	Nutrition

Total CarboHydrades	Nutrition
Total Saturated Fat	Nutrition
Vitamin A	Nutrition
Diameter	Measures
Gross weight	Measures
Net Weight	Measures
Non GTIN Pallet Height	Measures
Peg Horizontal	Measures
Percentage of Alcohol per Volume	Measures
Width	Measures
Product Strength	Measures
Depth	Measures
Drained Weight	Measures
Height	Measures
Non GTIN Pallet Gross Weight	Measures
Packaging Weight	Measures
Peg Vertical	Measures
Volume	Measures
Recommended Base Price for Unit	TM Specific

---

## Locale/Language Specific Configurations

To configure the language specific attributes, follow the steps explained in Configure Language Specific Attributes section in Chapter 3 – Configuration. After this, create a folder for the locale and copy the files within en\_US folder to the newly created folder. For example, if the locale is French, create a folder fr\_FR and then copy the files from en\_US to fr\_FR.

## Configure DataPool Catalog for supply side

The following table provides the configurations that need to be done in DataPoolCatalog in WebSphere Product Center. For more information on DataPool Catalog.

Key	Value	Description
00000778955540000077895554	145	Do not change this value.
MessageID	122	This is a unique ID sequence. This has to be reset to 0 for the first time and then onwards increases by one

Key	Value	Description
		for each message. This will be repended with a unique date-time combination.
CorrelationIDHeader	226	This is the same as MessageID.
PrincipalHeader	noname	This is the user ID for Transora.
7981315112271.8380160030003	105	This is the combination of Trading Partner GLN and the exchange GLN. These two are separated by a ".". The value is the conversation ID sequence.
ExchangeGLN	8380160030003	GLN of the respective exchange, this needs to be updated as per the target exchange GLN value
ExternalPublicationID	16	This is the unique ID sequence used for each publication message.

## Configure DataPool Catalog for demand side

The following table provides the configurations that need to be done in DataPoolCatalog in WebSphere Product Center.

Key	Value	Description
MessageID	0	This is a unique ID sequence. This has to be reset to 0 or any value that the user wants as the initial value. Consequently the value increases by one for each message. This will be prepended with a unique date-time combination.
UserID	xxxxx	This is the user – id to access the UCCnet data pool services
UserPWD	xxxxx	This is the password to access the UCCnet data pool services
uniqueCreatorIdentification	0	The unique creator identifier is a unique id that will identify each block in the same XML message that is sent to UCCnet data pool services. For example, while adding multiple trade items, each block will have a different unique creator identification id.  This value has to be reset to 0 or any value that the user wants as the initial value, for the first time and



Key	Value	Description
		consequently the value increases by one for each block in the same message.
ExchangeGLN	8380160030003	GLN of the respective exchange, this needs to be updated as per the target exchange GLN value.
SelfGLN	xxxxxx	This is the GLN of the trading partner who is using this application.
DefaultLocaleString	en_US	This is the default locale.

## ***IBM Global Data Synchronization for WebSphere Product Center Uninstallation***

This section provides the steps for uninstalling IBM Global Data Synchronization for WebSphere Product Center solution.

Uninstallation can be done in two ways.

- Using Command Line
- Using Installer

### **Using Command Line**

To uninstall the IBM Global Data Synchronization for WebSphere Product Center solution using the command line, type the following at the command prompt:

```
<GDS_installLocation>/_uninst/uninstaller.bin -silent
```

### **Using Installer**

To uninstall the IBM Global Data Synchronization for WebSphere Product Center solution using the installer:

1. Invoke the .bin executable from outside the \_uninst directory, by the following commands:

```
Export DISPLAY = <client display ip address>:0.0
./uninst/uninstallGDS.bin
```

**Note:** The uninstaller should be invoked from outside the \_uninst directory. No other shells should be using the \_uninst directory as their current working directory.

**Note:** If you need to move the installation of GDS onto another path, you must run the uninstaller and re-install GDS in the desired location. Do not move the application directory manually.

2. The *Splash* window is displayed for a few seconds and then the *Welcome* window is displayed. This provides the product name and version that you are uninstalling. Also displays the vendor name and the Website information.
3. Click **Next**. The *Summary Information* screen displays the directory location from which IBM Global Data Synchronization for WebSphere Product Center is uninstalled. It also provides the features that were selected to be uninstalled.
4. Click **Next**. The uninstallation process starts. Once the process is completed the *Uninstallation Complete* window is displayed.
5. Click **Finish**. This completes the uninstallation process.

---

## Ch 3 Configuration

This chapter details the configurations that you can make for the IBM Global Data Synchronization for WebSphere Product Center Demand solution, according to your requirement. These configurations need to be made in WebSphere Product Center version 5.1.0.3. The settings reflect in the IBM Global Data Synchronization for WebSphere Product Center solution.

Note: You need to have working knowledge of WebSphere Product Center version 5.1.0.3

Topics:

Configuration common for demand and supply

- [Modify or Move the Names of Attributes](#)
- [Modify or Move the Names of Attribute Groups](#)
- [Configure Target Markets](#)
- [Configure Language Specific Attributes](#)
- [Configure Trade Item Status](#)
- [Configure Enrichment Completion Criteria](#)
- [Configure Item Data Compliance Check](#)
- [Configure Approval Process](#)
- [Configure Classification Scheme](#)
- [Configure Users and Roles](#)
- [Configure Workflows](#)
- [Mapping Between Internal Category and the UDEX Category](#)

Other configurations while installing the supply side component

- [Configuring Interoperability Attributes](#)

For configuring, you need to log on to WebSphere Product Center version 5.1.0.3.






### ***Modify or Move the Names of Attributes***

You can change the names of global and global-local attributes that is displayed in the IBM Global Data Synchronization for WebSphere Product Center application.

### **Modify Global Attributes**

The Global Attributes of the data pool is mapped to the Global Catalog in WebSphere Product Center.






To change the names of Global Attributes:

1. Go to Product Manager>Catalogs>Catalog Console. You can also modify the spec from the Data Model Manager> Specs/Mappings>Specs Console.
2. Click Global\_Attributes\_Spec next to Global\_Catalog.
3. Click **Edit** to edit the Global\_Attribute\_Spec.
4. Click  next to Global\_Attributes\_Spec to add a new attribute.
5. To modify the names of the attributes, click on the attribute name.
6. To move an attribute, click on the  to move the attribute up or  to move the attribute down.
7. To add attributes, click  next to the attribute name for which you want to add a attribute.
8. To delete an attribute, click  for the attribute that you want to delete.
9. Click **Save** to save your changes.

## Modify Global-Local Attributes

The Global-local Attributes of the data pool is mapped to the Global-Local Catalog in WebSphere Product Center.

To change the names of Global-Local Attributes:

1. Go to Product Manager>Catalogs>Catalog Console. You can also modify the spec from the Data Model Manager> Specs/Mappings>Specs Console.
2. Click Global\_Local\_Attributes\_Spec next to Global\_Local\_Catalog.
3. Click **Edit** to edit the Global\_Local\_Attribute\_Spec.
4. Click  next to Global\_Local\_Attributes\_Spec to add a new attribute.
5. To modify the names of the attributes, click on the attribute name.
6. To move an attribute, click on the  to move the attribute up or  to move the attribute down.
7. To add attributes, click  next to the attribute name for which you want to add a attribute.
8. To delete an attribute, click  for the attribute that you want to delete.

## Modify or Move the Names of Attribute Groups

Multiple attributes can be arranged into a named group. Attribute groups are defined and named in one place so that they can be reused and easily updated. For example, in the IBM Global Data Synchronization for WebSphere Product Center solution, all attributes related to Logistics are grouped under the Logistics tab. The names of these groups can be modified.

These changes are done in WebSphere Product Center and they reflect in the IBM Global Data Synchronization for WebSphere Product Center solution.


To change the names of Attributes Groups:

1. Go to Data Model Manager>Attribute Collections>Attribute Collection Console.
2. Click on the attribute group that you want to modify. The list of specs associated with the group is displayed.
3. Under **Specs in Collection**, double-click on the spec. The list of attributes for that spec is displayed under **Attributes for Collection in Spec**.
4. To delete the attribute(s), select those attributes that you want to delete, and click **Remove selected attributes**.
5. To add attributes, click **Search**. The list of attributes is displayed.
6. Select the attributes that you want to add and click **Selected**. The selected attributes are added to the attribute collection.
7. Click **Save** to commit your changes.

## ***Configure Target Markets***

The list of target markets that are accessible in the IBM Global Data Synchronization for WebSphere Product Center solution is preset using WebSphere Product Center.

To setup the Target Markets:

1. Go to Product Manager>Lookup Tables> Lookup Table Console.
2. Select Target\_Market\_Lookup and click .
3. You can add new target markets, edit or remove existing target markets.
4. You can also enable or disable target markets.
5. Save your settings.

All these changes should reflect in the IBM Global Data Synchronization for WebSphere Product Center application.

## ***Configure Language Specific Attributes***

WebSphere Product Center provides the ability to localize attribute values for any catalog attribute or hierarchy node. Locales are created based on language/country pairs, which provide variances across countries (that is, U.S. English, British English). You can localize a single attribute value or display name.

To localize the Language Specific Attributes:



1. Go to Data Model Manager>Security> Company Attributes.
2. Select the language/country pairs supported.

3. You can also remove the supported languages and countries.
4. Click **Save** to save the spec.
5. Go to each spec, and enable the **Localized** option. The set of locales are displayed. Select the locale and click **Add**.
6. Click on the attribute to localize the attribute's display name.
7. Enable the Localized option under Attribute Collection Associations. The attribute is localized for all the locales.
8. Save your settings.
9. To view these localized attributes data from the UI perspective go to Home->My Settings and Add Locales from the Available Locales to Selected Locales and save the settings.

### ***Configure Trade Item Status***

A trade item goes through a number of statuses in the IBM Global Data Synchronization for WebSphere Product Center application. If the status display name is changed in WebSphere Product Center, the same will be reflected in IBM Global Data Synchronization for WebSphere Product Center UI.

To change the display name for the Trade Item Status:

1. Go to Product Manager>Lookup Tables> Lookup Table Console.
2. For Item\_Status\_Lookup and click .
3. Click on the status name or the **Edit** icon  to rename the status.
4. Click **Save** to commit the changes.

### ***Configure Enrichment Completion Criteria***

To configure the Enrichment Completion Criteria:

1. Go to Data Model Manager>Attribute Collections>Attribute Collection Console.
2. Click **Required**.
3. Under **Specs in Collection**, double-click on the spec. The list of attributes for that spec is displayed under **Attributes for Collection in Spec**.
4. To delete the attribute(s), select those attributes that you want to delete, and click **Remove selected attributes**.
5. To add attributes, click **Search**. The list of attributes is displayed.
6. Select the attributes that you want to add and click **Selected**. The selected attributes are added to the attribute collection.

7. Click **Save** to commit your changes.

## **Configure Approval Process**

The approval process for trade items, trade item links, and trading partners can be automated. This is done by setting the required approval process to ON or OFF in the `gds.properties` file.

To set the Approval Process:

1. At the command prompt, type `$TOP/etc/default/gds.properties`.
2. Change the settings to ON or OFF.

## **Configure Classification Scheme**

Classification scheme is a hierarchical structure of product categories and product attributes. Currently, IBM Global Data Synchronization for WebSphere Product Center supports the UDEX classification scheme.

The configuration can be done through the hierarchy mapping console or selecting the hierarchy from the left navigation pane.

To configure the Classification Scheme for supply side:

1. From the left navigation pane, select the hierarchy.
2. Select the category that you want to enable.
3. Select the check box for Enabled.

To configure the Classification Scheme for demand side

1. Go to hierarchy Mapping console
2. Map the required classification hierarchy to the UDEX classification.

## **Configure Users and Roles**

The roles are pre-seeded in the solution. The users need to be created. For more information, refer the *Appendix A – Pre-seeded Roles, Access Control Groups, and Selections for IBM Global Data Synchronization for WebSphere Product Center*.

To create a new user:

1. In the left navigation pane, from the drop-down menu, select Default Organization Hierarchy. The hierarchy is displayed under the drop-down menu.
2. Click on the plus sign to expand Default Organization Hierarchy.
3. Point the cursor on Default Organization. Right-click and select **Add User**.
4. Fill in the fields under User Profile.
5. Specify the password for this user.
6. From the list of roles that are displayed, select a role for this user.

7. Click **Save**.
8. Refresh the left navigation pane.
9. Click on view users. The *User Console* displays the newly created user.
10. Enable the new user.

## Configure Workflows

During solution setup, you have the option to turn on/off the global attributes enrichment process, variant attribute enrichment process, and the approval process. The following list of approval process can be turned on/off. If any of the approval process is turned OFF, then it is automated within the solution.

- New Item Approval
- Modify Item Approval
- Delete Item Approval
- New Link Approval
- Delete Link Approval
- Modify Link Approval
- New Trading Partner Approval
- Modify Trading Partner Approval
- Delete Trading Partner Approval

To configure the workflows:

1. Go to `$TOP/etc/default/gds.properties` file using any editor.
2. For configuring the global enrichment, set `global_enrichment` to true or false. Similarly for switching the variant enrichment, set `local_enrichment` to true or false.
3. For configuring the approval processes, set the respective process to ON or OFF.
4. Restart the server.

Note: This configuration can be done when installing the solution using the Install Shield.

## Configure Information Provider

The list of information providers that are accessible in the IBM Global Data Synchronization for WebSphere Product Center solution is preset using WebSphere Product Center.

To configure the Information Providers:

1. In the left navigation pane, from the drop-down menu, select `Trading_Partner_Catalog`. The hierarchy is displayed under the drop-down menu.
2. Right-click on the country and click **Add Item**. The details page for the trading partner is displayed.




3. Specify the following:
  - a. In the **Name** field, type the name for the Information Provider.
  - b. In the **GLN** field, type the Global Location Number.
  - c. In the **GLN Identifier** field, select any of the available options. Options available are Manufacturer, Exchange, Retailer, and Supplier.
  - d. In the **Country** field, type the country of the Information Provider.
  - e. Specify the status as Approved and the TP status as Active.
  - f. Specify the item type as Information Provider.
4. Click **Save** to save the settings.

### **Configure Trade Item Data Compliance Check**

Once the trade item has been enriched, it should be submitted for a compliance check. The validation rules for this compliance check are setup using WebSphere Product Center.

To configure Trade Item Data Compliance Check:

1. Go to Data Model Manager>Scriptings>Scripts Console.
2. From the drop-down menu, select Distribution Script. The *Scripts Console* is displayed.
3. Click on the **Edit** icon  next to Trade Item Link Validation.
4. Edit the script and click **Save** to commit the changes.

**Note:** When creating selections, ensure that the Information Provider that is configured is selected. For more information, refer Create Selections in *Chapter 4 – Administration*.

### **Mapping Between Internal Category and the UDEX Category**

Internal Category is the categories maintained by the organization. UDEX-UK category is the category in the UDEX UK data pool.

To map the attributes between the Internal Category and the UDEX Category:

1. Go to Product Manager>Hierarchy Maps>Hierarchy Mapping Console.
2. You can create a new map or modify an existing map.
3. Click New to create a new map.
4. Select the Internal Category and then the UDEX category. This maps the selected internal category to the UDEX category.
5. The settings are saved automatically.

Ensure that mapping to UDEX Categories are done only for the Categories accessible by a user. Also, the field "udex\_to\_internal\_hierarchy\_mapping\_without\_security\_breach" should be set to true in the `gds.properties` file. If this is not set to true, there will be a performance penalty.

### ***Configure Interoperability Attributes- (only supply side)***

Categories are defined for different data pools. Currently, UDEX-UK is supported. To configure the Interoperability Attributes, open the `gds.properties` file located in `etc/default/` directory. Set the value for `Interoperability_UDEX_UK` to `True`. This enables you to specify the Interoperability attribute values in the IBM Global Data Synchronization for WebSphere Product Center.

Note: All the UDEX-UK interoperability attributes are not preseeded. For attributes that are not preseeded, refer Transora UDEX-UK interoperability.

To configure the Interoperability Attributes that is not preseeded:

1. In the left navigation pane, from the drop-down menu, select the data pool for which you want to configure the attributes. The list of categories for the selected data pool is displayed.
2. Click on a category. The details for the category are displayed on the right side.
3. If the specs are pre-seeded, edit the spec.
4. If there are no specs, you need to create a new spec.
5. Once the spec is created, add it to the attribute collection.



---

## Ch 4 Administration

This chapter provides information about security and user management in IBM Global Data Synchronization for WebSphere Product Center solution. WebSphere Product Center 5.1.0.3 is used for setting the security and the roles for the various users.

Topics:

- [Security](#)
- [Security Steps](#)

### **Security**

Security requirements for IBM Global Data Synchronization for WebSphere Product Center are divided into two parts. They are:

- Functional Security
- Data Security

### **Functional Security**

Functional Security controls the actions that can be performed by the user on the System. Depending on the privileges granted, the user will be able to perform a limited set of actions.

### **Data Security**

Data Security is implemented using the Selections feature in WebSphere Product Center. Security is provided for the Trade Items, Trade Item Links, Trading Partners, Target Markets and Transactions.

### **Security Steps**

The steps to set the security are:

1. Log on to WebSphere Product Center
2. Create a Role
3. Create an Access Control Group (ACG)
4. Create Selections
5. Provide Data Access Privileges
6. Create User

## Create a Role

The role is one of the central concepts of the IBM Global Data Synchronization for WebSphere Product Center solution. A role is a collection of tasks, services, and information for a user or a group of users. The role defines which service and information the user should be able to access and the activities that can be performed. The role also defines the visualization of the contents and the navigation structure within the solution.

Roles are stored in the WebSphere Product Center database. For information about the roles pre-seeded in the solution, refer *Appendix – Pre-seeded Roles and Selections for IBM Global Data Synchronization for WebSphere Product Center*.

Alternatively, you can create new roles. To do this

1. Log on to WebSphere Product Center.
2. Go to Data Model Manager > Security > Role Console.
3. Click **New**.
4. Type the name for the role.
5. From the drop-down menu select default ACG and provide the access privileges for this role.
6. Save the settings.

The roles and Access Control Groups (ACG) that are set in WebSphere Product Center are applied in IBM Global Data Synchronization for WebSphere Product Center.

## Create an Access Control Group (ACG)

The Access Control Group is used to control the access levels for each role. Permissions are set to define the type of access granted to the role and the associated entity. When you set up permissions, you specify the level of access for the role and the entity. To specify access rights, you identify them in access control groups.

The ability to control a user's access to view, edit, and delete entities can be restricted based on the access privilege definition of the role that is assigned to the user. To restrict access to an entity, the role defined with restricted access privileges must be associated with the ACG that the entity is using.

Therefore, a single selection can be made available to a specific ACG and all roles that are part of the ACG will have access to that entity. A user is allowed access to an entity once they are assigned to the role.

There are two types of Access Control Groups. They are:

- [UDEX CAT SPECIFIC] ACG – This is the UDEX category specific access control group. The available roles can be:
  - [UDEX CAT SPECIFIC] Global Brand Manager
  - [UDEX CAT SPECIFIC] Global Product Executive

- [UDEX CAT - TM SPECIFIC] Brand Manager
- [[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
- [[UDEX CAT - TM - ATTR GROUP SPECIFIC] User
- [UDEX CAT - TM SPECIFIC] Variant Publisher
- [TM Specific] ACG – This is the target market specific access control group. The available roles can be:
  - [UDEX CAT - TM SPECIFIC] Brand Manager
  - [[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
  - [[UDEX CAT - TM - ATTR GROUP SPECIFIC] User
  - [UDEX CAT - TM SPECIFIC] Variant Publisher
  - [TM Specific] Marketing Manager

You can create new ACGs. To create new ACGs:

1. Log on to WebSphere Product Center.
2. Go to Data Model Manager > Security > Access Control Groups > Access Control Group Console.
3. Click New.
4. Type the name for the ACG. From the Role drop-down menu, select the role that is created in step2.
5. Provide access privileges for the default ACG. Select the checkboxes for Catalog list, Hierarchy list, Selection list, and Selection Measures list.
6. Save the settings.

## Create Selections

The created ACGs are linked to Selections that are created through WebSphere Product Center. These Selections are used to achieve Category and Target Market based security.

When a Selection is created, it is associated with an ACG, a catalog, and a hierarchy. So, for achieving category-level security, a Selection is created for [UDEX CAT SPECIFIC] ACG, Global\_Catalog and UDEX\_Hierarchy\_UCCNet hierarchy. Similarly, for Target Market based security, Selection is created for [TM Specific] ACG, Global\_Local\_Catalog and Target\_Market hierarchy.

Once the Selection is created, the required Categories are selected. Once this is done, the users associated with the ACGs will have visibility into only those categories/ target markets that are selected in the Selection.

**Note:** All Categories (Internal and UDEX) that are accessible through selections must be enabled. If that is not done, the pagination will not be proper, that is the result size may not correspond to the defined page size.

The following selections need to be done:

1. Global Catalog Selection
2. Global Local Catalog Selection
3. Trading Partner Catalog Selection
4. GLN Selection
5. Filter Selection
6. Internal Hierarchy Selection
7. UCCNet Message Archive Catalog Selection

For information about pre-seeded Selections, refer *Appendix – Pre-seeded Roles and Selections for IBM Global Data Synchronization for WebSphere Product Center*.

### **Selection configurations common to both Demand and Supply**

#### **Global Catalog Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.
2. Select Global Catalog from the drop-down.
3. Select the version.
4. Select the Access Control Group.
5. Select the hierarchy as UDEX\_Hierarchy\_UCCNet. All the categories under Global Catalog Hierarchy are displayed.
6. Type a name for the selection and then select the categories.
7. Click Save to save the selection.

#### **Global Local Catalog Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.
2. Select the version.
3. Select the Access Control Group.
4. Select the hierarchy as Target\_Market\_Hierarchy. All the target markets under Global Local Catalog Hierarchy are displayed.
5. Type a name for the selection and then select the categories.
6. Click Save to save the selection.

#### **Trading Partner Catalog Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.

2. Select Trading Partner Catalog from the drop-down. Select the version.
3. Select the Access Control Group.
4. Select the hierarchy as Target\_Market\_Hierarchy. All the target markets under Target\_Market\_Hierarchy are displayed.
5. Type a name for the selection and then select the target markets.
6. Click Save to save the selection.

### **GLN Catalog Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.
2. Select <GLN catalog> from the drop-down.
3. Select the version.
4. Select the Access Control Group.
5. Select the hierarchy as IP\_Hierarchy. All the target markets under IP\_Hierarchy are displayed.
6. Type a name for the selection and then select the information providers.
7. Click Save to save the selection.

### **Demand-side specific selections**

In addition to the above selection configurations, for demand side the following selection configurations need to be done in WebSphere Product Center 5.1.0.3

#### **Filter Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.
2. Select Filter Catalog from the drop-down.
3. Select the version.
4. Select the Access Control Group.
5. Select the hierarchy as Default Filter Hierarchy. All the categories under Filter Catalog are displayed.
6. Type a name for the selection and then select the categories.
7. Click Save to save the selection.

#### **Internal Hierarchy Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.
2. Select Global Catalog from the drop-down.
3. Select the version.



4. Select the Access Control Group.
5. Select the hierarchy as Internal Hierarchy. All the categories under Global Catalog Hierarchy are displayed.
6. Type a name for the selection and then select the categories.
7. Click Save to save the selection.

### **UCCNet Message Archive Catalog Selection**

1. Go to Product Manager > Selections > Selection Console. Click New Basic Selection.
2. Select UCCNet Message Archive Catalog from the drop-down.
3. Select the version.
4. Select the Access Control Group.
5. Select the hierarchy as UCCNetMessageArchiveHierarchy. All the categories under UCCNet Message Archive Catalog Hierarchy are displayed.
6. Type a name for the selection and then select all the categories.
7. Click Save to save the selection.

### **Provide Data Access Privileges**

Go to Security > Access Privileges > Catalog Access Console. Provide access privileges to the attribute groups for the Global Catalog and the Global Local Catalog. You can either make the attribute groups viewable or editable.

### **Create User**

For information about creating the user, refer the section Configure Users and Roles in *Chapter 3 – Configuration*.

---

## Ch 5 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. Also provided are sample scripts which show how the scenarios are implemented.

**Note:** IBM Global Data Synchronization for WebSphere Product Center does not have separate import/export module. For release 1.1.1, WebSphere Product Center 5.1.0.3 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.

The exchange compliance validation script needs to be called once the data is imported (to ensure data pool compliance).

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

### Sample XML

This sample xml file has just the mandatory attributes that is required to create a trade item.

```
<TrigoXML>

<Header>
  <Transaction_Id></Transaction_Id>
  <File_Id>01</File_Id>
  <Record_Id>01</Record_Id>
  <Creation_Date>20040428</Creation_Date>
  <Creation_Time>154015</Creation_Time>
</Header>

<Item>
  <Attribs>

    <GlobalAttribs>

      <GlobalTradeItemNumber>72230000000008</GlobalTradeItemNumber>
      <GTINName>CHILD</GTINName>
      <ProductType>EACH</ProductType>
      <ItemStatus>Approved</ItemStatus>

      <BrandName>Coke</BrandName>
      <Pack>8</Pack>

      <Classification>

        <GlobalClassificationCode>000000001.000000002.000000012</GlobalClassif
icationCode>
        <GlobalClassificationDescription>ALCOHOL BASED
COCKTAILS/MIXES</GlobalClassificationDescription>

        <InternalClassificationCode>10000056</InternalClassificationCode>
        <InternalClassificationDescription>Alcohol</InternalClassificationDesc
ription>
      </Classification>
    </GlobalAttribs>

    <LocalAttribs>
      <TargetMarket>US</TargetMarket>
      <InformationProvider>7981315112271</InformationProvider>
    </LocalAttribs>
  </Attribs>
</Item>
</TrigoXML>
```

```

        <InformationProviderName>TEST_GLN</InformationProvider>
        <ManufacturerGLN>US</ManufacturerGLN>
        <IsPrivate>true</IsPrivate>
        <StartAvailabilityDate>30-AUG-2004
00:00</StartAvailabilityDate>
        <MeasurementCharacteristics>

            <Height>
                <UOM>IN</UOM>
                <Value>30</Value>
            </Height>

            <Width>
                <UOM>IN</UOM>
                <Value>31</Value>
            </Width>

            <Depth>
                <UOM>IN</UOM>
                <Value>31</Value>
            </Depth>

        </MeasurementCharacteristics>

        <OrderingUnitIndicator>true</OrderingUnitIndicator>
    </LocalAttribs>

    </Attribs>
</Item>
</TrigoXML>

```

## 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(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.

3. `HashMap TradeItem::validateTradeItemAttributeGroup(String sAttributeGroup)`

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.

## Scenarios and Sample Scripts

This section lists down the various customer scenarios for import and export and also provides sample scripts for those scenarios.

Note: For information about how to import or export trade items, links, and trading partners, refer the *WebSphere Product Center User's Guide*.

### For Import

The various scenarios for import are:

- [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](#)
- [Auto-update the Status of a Trade Item after Import](#)
- [Import the Trade Item and Ensure that it is Compliant with a Particular Exchange](#)
- [Import Trade Item Links in a Particular Status](#)
- [Updating Trade Item Links through Import](#)
- [Auto-update the Status of Trade Item Link after Import](#)
- [Import Trading Partners](#)
- [Import Trading Partners in a Particular Status](#)
- [Auto Update the Status of the Trading Partner after Import](#)
- [Catalog to Catalog Import](#)
- [Notifications after 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.

```
logFile = createOtherOut("GDS Sample Import - Import Global Items Only.txt");

var sGTIN = "";
var sGTINName = "";
var sProductType = "";
var sInternalCategoryCode = "";
var sDataPoolCategoryCode = "";
var status = "";

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
Import Global Items Only/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);
var hmAttributePathsToValues = [];

var specMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
if(specMap != null)
{
    var hmDstPathsToSrcPaths = [];
    var hmDstPathsToSrcPaths = specMap.getMappings();

    forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
    {
        if(dstPath.contains("GlobalTradeItemNumber"))
        {
            sGTIN = parseXMLNode(srcPath.getRidOfRootName());
            logFile.writeln(dstPath+" "+sGTIN);
        }
        else if(dstPath.contains("GTINName"))
        {
            sGTINName =
parseXMLNode(srcPath.getRidOfRootName());
            logFile.writeln(dstPath+" "+sGTINName);
        }
        else if(dstPath.contains("ProductType"))
        {
            sProductType =
parseXMLNode(srcPath.getRidOfRootName());
            logFile.writeln(dstPath+" "+sProductType);
        }
        else if(dstPath.contains("GlobalClassificationCode"))
        {
            sDataPoolCategoryCode =
parseXMLNode(srcPath.getRidOfRootName());
            logFile.writeln(dstPath+" "+sDataPoolCategoryCode);
        }
    }
}
```

```

        }
        else
        {
            var value =
checkString(parseXMLNode(srcPath.getRidOfRootName()), "");
            logFile.writeln(dstPath+", "+value);
            if(value != "")
            {
                hmAttributePathsToValues[dstPath] = value;
                out.writeln(dstPath+" =====> "+value);
            }
        }
    }

    sInternalCategoryCode =
parseXMLNode("Item/Attribs/GlobalAttribs/Classification/InternalClassificationCode");
    logFile.writeln("InternalCategoryCode, "+sInternalCategoryCode);

    if(sGTIN != ""
        && sGTINName != ""
        && sProductType != ""
        && sInternalCategoryCode != ""
        && sDataPoolCategoryCode != ""
    )
    {
        tradeItem = new
TradeItemSS(sGTIN, sGTINName, sProductType, sInternalCategoryCode, sDataPoolCategoryCode);
        if(hmAttributePathsToValues != null &&
hmAttributePathsToValues.size() != 0)
        {
            out.writeln("setting attributes
"+hmAttributePathsToValues);

            tradeItem.setTradeItemAttributesSS(hmAttributePathsToValues);
        }

        tradeItem.saveTradeItemSS();

        createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "master
");

        createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "master
r");

        out.writeln("=====");
        out.writeln("Global Item Saved");
        out.writeln("=====");
        out.writeln(tradeItem);

        out.writeln("=====");
        out.writeln("Creating Transaction");
        var bCreateTxn =
createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "master");
        // out.writeln((bCreateTxn == true) ? "Done":"Failure");
        out.writeln("=====");

        out.writeln("=====");
        out.writeln("Creating Notification");

```

```

        var bCreateNotf =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_COMPLETED","master");
//        out.writeln((bCreateNotf == true) ? "Done":"Failure");
        out.writeln("=====");
    }

}

logFile.save("GDS Sample Import - Import Global Items Only.txt");

```

## Import Global and Local Trade Items

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// This script inserts a trade item (a global item and its variant) into GDS
in any given status
//
// Assumptions: -
// *****
// 1. Both Global Item and Local Item are being imported.
// 2. The GDS Item Status is mentioned in the Sample XML file.
// 3. The Notification Type is set to Approved while creating Transaction and
Notification. This can be set to anything.
// 4. The XML file is taken from the import and it is always the latest file.
The name of the Import is hardcoded here.
// 5. There are two spec maps created in the system, called,
//     a) GDS XML Global Attributes Spec Map (this maps the Mandatory Global
Attributes to the Sample XML file ).
//     b) GDS XML Local Attributes Spec Map (this maps the Mandatory Local
Attributes to the Sample XML file ).
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function removeTradeItem(tradeItem, sGTIN)
{
    logFile.writeln("Performing Cleanup.....");
    var items = [];
    items = tradeItem.getItems();

    if(items != null && items.size() != 0)
    {
        var globalItem = items[0];
        var localItem = items[1];
        var gtin_tm_ip = "";

        if(globalItem != null)
        {
            gtin_tm_ip = globalItem.getCtgItemPrimaryKey()+"_XX_XX";
            deleteCtgItem(globalItem);
            logFile.writeln("Global Item "+gtin_tm_ip+" deleted from
Global Catalog");
        }
        if(localItem != null)
        {
            deleteCtgItem(localItem);
            logFile.writeln("Local Item "+gtin_tm_ip+" deleted from
Local Catalog");
        }

        var tradeCtg = getCtgByName("TradeCatalog");
        var tradeCtgItem = tradeCtg.getCtgItemByPrimaryKey(gtin_tm_ip);
    }
}

```



```

        if(tradeCtgItem == null)
        {
            var tradeCtgItemSet =
tradeCtg.getCtgItemByAttributeValue("TradeCatalogPrimarySpec/GTIN",sGTIN);
            if(tradeCtgItemSet != null)
            {
                foreachItemSetElement(tradeCtgItemSet,item)
                {
                    tradeCtgItem = item;
                }
            }
        }

        if(tradeCtgItem != null)
        {
            deleteCtgItem(tradeCtgItem);
            logFile.WriteLine("TradeItem "+gtin_tm_ip+" deleted from
Trade Catalog");
        }
    }

    // Now remove the category that was created in GTIN_Hierarchy

    var gtinTree = getCategoryTreeByName("GTIN_Hierarchy");
    var gtinCatSet = gtinTree.getCategorySetByPrimaryKey(sGTIN);
    if(gtinCatSet != null && gtinCatSet.getCategorySetSize() != 0)
    {
        foreachCategorySetElement(gtinCatSet,gtinCat)
        {
            gtinCat.deleteCategory();
            logFile.WriteLine("Category "+sGTIN+" deleted");
            gtinTree.saveCategoryTree();
        }
    }

    logFile.WriteLine("Cleanup complete....");
}

var logFile = createOtherOut("GDS Sample Import - New TradeItems.txt");
var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
New TradeItems/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

var globalSpecMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
var localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");
catchError(oErr)
{
    foreachXMLNode("Item/")
    {
        var statusFromXML = "";
        var bValueSetForLocal = false;
        var hmGlobalAttributePathsToValues = [];
        var hmLocalAttributePathsToValues = [];

        if(globalSpecMap != null)

```

```

        {
            var sGTIN = "";
            var sGTINName = "";
            var sProductType = "";
            var sInternalCategoryCode = "";
            var sDataPoolCategoryCode = "";

            var hmDstPathsToSrcPaths = [];
            var hmDstPathsToSrcPaths = globalSpecMap.getMappings();

            forEachHmElement(hmDstPathsToSrcPaths, dstPath,
srcPath)
            {
                if(dstPath.endsWith("GlobalTradeItemNumber"))
                {
                    sGTIN =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
                    logFile.writeln(dstPath+", "+sGTIN);
                }
                else if(dstPath.contains("GTINName"))
                {
                    sGTINName =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
                    logFile.writeln(dstPath+", "+sGTINName);
                }
                else if(dstPath.endsWith("ProductType"))
                {
                    sProductType =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
                    logFile.writeln(dstPath+", "+sProductType);
                }
                else
                if(dstPath.endsWith("GlobalClassificationCode"))
                {
                    sDataPoolCategoryCode =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
                    logFile.writeln(dstPath+", "+sDataPoolCategoryCode);
                }
                else if(dstPath.endsWith("GDSItemStatus"))
                {
                    var globalStatusCtg =
getCtgByName("Item_Status_LookUp");
                    statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                    if(statusFromXML != "" &&
globalStatusCtg.containsByPrimaryKey(statusFromXML))
                    {
                        hmGlobalAttributePathsToValues[dstPath] = statusFromXML;
                    }
                    else
                    {
                        hmGlobalAttributePathsToValues[dstPath] = "Draft";
                    }
                }
                else
                {

```

```

        var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
        logFile.writeln(dstPath+", "+value);
        if(value != "")
        {

            hmGlobalAttributePathsToValues[dstPath] = value;

            logFile.writeln(dstPath+", "+value);
        }
    }

    sInternalCategoryCode =
parseXMLNode("Attribs/GlobalAttribs/Classification/InternalClassificationCode
");

    logFile.writeln("InternalCategoryCode, "+sInternalCategoryCode);

    if(sGTIN != ""
        && sGTINName != ""
        && sProductType != ""
        && sInternalCategoryCode != ""
        && sDataPoolCategoryCode != ""
    )
    {
        tradeItem = new
TradeItemSS(sGTIN, sGTINName, sProductType, sInternalCategoryCode, sDataPoolCateg
oryCode);

        if(tradeItem != null &&
(hmGlobalAttributePathsToValues != null &&
hmGlobalAttributePathsToValues.size() != 0))
        {
            logFile.writeln("Setting attributes for
Global Item "+hmGlobalAttributePathsToValues);

            tradeItem.setTradeItemAttributesSS(hmGlobalAttributePathsToValues);
        }
    }

    // Now set the Local Item

    if(localSpecMap != null)
    {
        hmDstPathsToSrcPaths = [];
        hmDstPathsToSrcPaths = localSpecMap.getMappings();

        forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
        {
            if(dstPath.endsWith("GDSItemStatus"))
            {
                var localStatusCtg =
getCtgByName("Item_Status_Local_LookUp");
                statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

                if(statusFromXML != "" &&
localStatusCtg.containsByPrimaryKey(statusFromXML))
                {

```

```

        hmLocalAttributePathsToValues[dstPath] = statusFromXML;
    }
    else
    {
        hmLocalAttributePathsToValues[dstPath] = "Draft";
    }
}
else
{
    var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
    if(value != "")
    {
        hmLocalAttributePathsToValues[dstPath] = value;
        logFile.writeln(dstPath+", "+value);

        if(!dstPath.contains("GlobalTradeItemNumber"))
        {
            bValueSetForLocal = true;
        }
    }
}

if(bValueSetForLocal == true)
{
    if(tradeItem != null &&
(hmLocalAttributePathsToValues != null &&
hmLocalAttributePathsToValues.size() != 0))
    {
        logFile.writeln("Setting attributes for
Local Item "+hmLocalAttributePathsToValues);

        tradeItem.setTradeItemAttributesSS(hmLocalAttributePathsToValues);

        if(checkString(hmGlobalAttributePathsToValues["Global_Attributes_Spec/
GDSItemStatus"], "") == "Draft")
        {
            tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemStatu
s", "Draft With Variant");
            logFile.writeln("Updating Global
status to Draft with Variant since local Item is being inserted");
        }
    }
}

var tradeItemSaved = false;

if(tradeItem != null)
{
    catchError(saveError)
    {
        tradeItemSaved = tradeItem.saveTradeItemSS();
    }
    if(saveError != null || tradeItemSaved == false)
    {

```

```

        logFile.writeln("There was an error/failure while
saving the tradeitem....");
        logFile.writeln(""+saveError);
        logFile.writeln("Now rolling back the
changes....");

        removeTradeItem(tradeItem,tradeItem.getTradeItemAttributeValueSS("Glob
al_Attributes_Spec/GlobalTradeItemNumber"));
    }
    else
    {

        logFile.writeln("=====");
        logFile.writeln("Trade Item Saved");

        logFile.writeln("=====");
        logFile.writeln(tradeItem);

        logFile.writeln("=====");
        logFile.writeln("Creating Transaction");
        var bCreateTxn =
createTransaction(tradeItem,"NEW_ITEM_APPROVAL_PENDING","master");

        logFile.writeln("====="+bCreateTxn+"=====");

        logFile.writeln("=====");
        logFile.writeln("Creating Notification");
        var bCreateNotf =
createNotification(tradeItem,"NEW_ITEM_APPROVAL_PENDING","master");

        logFile.writeln("====="+bCreateNotf+"=====");

    }
    }
    else
    {
        logFile.writeln("TradeItem is null");
    }
}

logFile.save("GDS Sample Import - New TradeItems.txt");
}
if(oErr != null)
{
    logFile.writeln("ERROR!! "+oErr);
    logFile.save("GDS Sample Import - New TradeItems.txt");
}
}

```

## Import Trade Item in a Particular Status

```
////////////////////////////////////
////////////////////////////////////
// This script updates a trade item into GDS in any given status
//
// Assumptions: -
// *****
// 1. The GDS Item Status is mentioned in the Sample XML file.
// 2. The Notification Type is set to Approved while creating Transaction and
Notification. This can be set to anything.
// But it should correspond to the Item status given in the XML file.
// 3. The XML file is taken from the import and it is always the latest file.
The name of the Import is hardcoded here.
// 4. There are two spec maps created in the system, called,
// a) GDS XML Global Attributes Spec Map (this maps the Mandatory Global
Attributes to the Sample XML file ).
// b) GDS XML Local Attributes Spec Map (this maps the Mandatory Local
Attributes to the Sample XML file ).
////////////////////////////////////
////////////////////////////////////

logFile = createOtherOut("GDS Sample Import - Update TradeItems");

var sGTIN = "";
var STM = "";
var sIP = "";
var localStatusFound = false;

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
Update TradeItems/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

var globalSpecMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
var localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");

forEachXMLNode("Item/")
{
    var hmGlobalAttributePathsToValues = [];
    var hmLocalAttributePathsToValues = [];
    var bCheckGlobalStatus = false;

    if(globalSpecMap != null)
    {
        var hmDstPathsToSrcPaths = [];
        var hmDstPathsToSrcPaths = globalSpecMap.getMappings();

        forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
        {
            if(!dstPath.endsWith("ProductType"))
            {
                if(dstPath.endsWith("GlobalTradeItemNumber"))
                {
                    sGTIN =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
logFile.writeln(dstPath+", "+sGTIN);

```

```

        }
        else if(dstPath.endsWith("GDSItemStatus"))
        {
            var globalStatusCtg =
getCtgByName("Item_Status_LookUp");
            statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

            if(statusFromXML != "" &&
globalStatusCtg.containsByPrimarykey(statusFromXML))
            {
                hmGlobalAttributePathsToValues[dstPath] = statusFromXML;
            }
        }
        else
        {
            var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            logFile.println(dstPath+", "+value);
            if(value != "")
            {
                hmGlobalAttributePathsToValues[dstPath] = value;
                logFile.println(dstPath+", "+value);
            }
        }
    }
}

// Now set local attributes

localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");

if(localSpecMap != null)
{
    hmDstPathsToSrcPaths = [];
    hmDstPathsToSrcPaths = localSpecMap.getMappings();

    forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
    {
        if(dstPath.endsWith("TargetMarket"))
        {
            sTM =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "XX")
;
            hmLocalAttributePathsToValues[dstPath] = sTM;
            logFile.println(dstPath+", "+sTM);
        }
        else if(dstPath.endsWith("InformationProvider"))
        {
            sIP =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "XX")
;
            hmLocalAttributePathsToValues[dstPath] = sIP;
            logFile.println(dstPath+", "+sIP);
        }
        else if(dstPath.endsWith("GDSItemStatus"))
        {
            var localStatusCtg =
getCtgByName("Item_Status_Local_LookUp");

```

```

        statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

        if(statusFromXML != "" &&
localStatusCtg.containsByPrimaryKey(statusFromXML))
        {
            hmLocalAttributePathsToValues[dstPath] =
statusFromXML;

            bCheckGlobalStatus = true;

        }
        else
        {
            var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            if(value != "")
            {
                hmLocalAttributePathsToValues[dstPath] =
value;

                logFile.println(dstPath+", "+value);

            }
        }
    }

    if(sGTIN != "")
    {
        tradeItem = getTradeItemSS(sGTIN, sTM, sIP);
        if(tradeItem != null)
        {

            if(bCheckGlobalStatus == true)
            {
                var iItems = tradeItem.getItems();

                if(iItems != null && iItems.size() != 0)
                {
                    if(iItems[0] != null && iItems[1] == null)
// That means the local item doesnt exist and we are trying to create one.
                    {

                        if(checkString(tradeItem.getTradeItemAttributeValueSS("Global_Attributes_Spec/GDSItemStatus"), "") == "Draft")
                        {

                            tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemStatus", "Draft With Variant");

                            tradeItem.saveTradeItemSS();
                            logFile.println("Updating
Global status to Draft with Variant since local Item is being set");

                        }

                    }

                }

            }

            if(hmGlobalAttributePathsToValues != null &&
hmGlobalAttributePathsToValues.size() != 0)
            {
                logFile.println("setting global attributes
"+hmGlobalAttributePathsToValues);

                tradeItem.setTradeItemAttributesSS(hmGlobalAttributePathsToValues);

            }
        }
    }

```



```

        if(hmLocalAttributePathsToValues != null &&
hmLocalAttributePathsToValues.size() != 0)
        {
            logFile.writeln("setting local attributes
"+hmLocalAttributePathsToValues);

            tradeItem.setTradeItemAttributesSS(hmLocalAttributePathsToValues);
        }

        var tradeItemSaved = false;
        catchError(saveError)
        {
            tradeItemSaved = tradeItem.saveTradeItemSS();
        }
        if(saveError != null || tradeItemSaved == false)
        {
            logFile.writeln("There was an error/failure while
saving the tradeitem...");
        }
        else
        {

            logFile.writeln("=====");
            logFile.writeln("Item Saved");

            logFile.writeln("====="+tradeItemSaved+"=====
");
            logFile.writeln(tradeItem);

            logFile.writeln("=====");
            logFile.writeln("Creating Transaction");
            var bCreateTxn =
createTransaction(tradeItem,"MODIFY_ITEM_APPROVED","master");
            // logFile.writeln((bCreateTxn == true)
? "Done":"Failure");

            logFile.writeln("====="+bCreateTxn+"=====
=====");

            logFile.writeln("=====");
            logFile.writeln("Creating Notification");
            var bCreateNotf =
createNotification(tradeItem,"MODIFY_ITEM_APPROVED","master");
            // logFile.writeln((bCreateNotf == true)
? "Done":"Failure");

            logFile.writeln("====="+bCreateNotf+"=====
=====");
        }
    }
}

logFile.save("GDS Sample Import - Update TradeItems");

```

## Update an Already Existing Trade Item through Import

Refer the scripts for Import an item in a particular status and Import an item through auto status

## Auto-update the Status of the Trade Item after Import

Trade items can be updated with the auto status functionality. It means that if a trade item is being updated and the status is not set in the xml file, the script can correctly set the status of the trade item based on its existing status in the system.

```
////////////////////////////////////  
////////////////////////////////////  
// This script updates a trade item (a global item and its variant) into GDS  
// in any given status  
//  
// Assumptions: -  
// *****  
// 1. Both Global Item and Local Item are being imported.  
// 2. The GDS Item Status is mentioned in the Sample XML file and it is Local  
// Item status.  
// 3. The Notification Type is set to Approved while creating Transaction and  
// Notification. This can be set to anything.  
// But it should correspond to the Item status given in the XML file. In  
// this sample scenario,  
// we have set the local item status to Approved and therefore we are  
// setting the Notification Type to NEW_ITEM_APPROVED.  
// 4. The XML file is taken from the import and it is always the latest file.  
// The name of the Import is hardcoded here.  
// 5. There are two spec maps created in the system, called,  
// a) GDS XML Global Attributes Spec Map (this maps the Mandatory Global  
// Attributes to the Sample XML file ).  
// b) GDS XML Local Attributes Spec Map (this maps the Mandatory Local  
// Attributes to the Sample XML file ).  
////////////////////////////////////  
////////////////////////////////////  
  
function getToStatus(strStatus,checkCompliance)  
{  
    var setStatus = "";  
    var tradeItemStatusSet =  
tradeItemStatusLkp.getCtgItemByAttributeValue("TradeItem_AutoStatus_LookUp_Sp  
ec/From_Status",strStatus);  
    if(tradeItemStatusSet != null)  
    {  
        foreachItemSetElement(tradeItemStatusSet, statusItem)  
        {  
            if(checkCompliance == true)  
            {  
  
                if(checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookU  
p_Spec/Compliance_Check_Required"), "") == "YES")  
                {  
                    var toStatus =  
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp_Spec/To_  
Status"), "");  
  
                    if(toStatus != "")  
                    {  
                        setStatus = toStatus;  
                    }  
                }  
            }  
        }  
    }  
}
```

```

                                notif =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp_Spec/Not
ification"), "");
                                txn =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp_Spec/Tra
nsaction"), "");
                                break;
                                }
                                }
                                }
                                else
                                {
                                if(checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp
p_Spec/Compliance_Check_Required"), "") == "NO")
                                {
                                var toStatus =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp_Spec/To_
Status"), "");
                                if(toStatus != "")
                                {
                                setStatus = toStatus;
                                notif =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp_Spec/Not
ification"), "");
                                txn =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_LookUp_Spec/Tra
nsaction"), "");
                                break;
                                }
                                }
                                }
                                }
                                }
                                return setStatus;
}

catchError(oError)
{
logFile = createOtherOut("GDS Sample Import - Update Global And Local Items -
Auto Status");
var tradeItemStatusLkp = getCtgByName("TradeItem_AutoStatus_Lookup");
var notif = "";
var txn = "";

var sGTIN = "";
var sTM = "";
var sIP = "";
var globalAttributesModified = false;
var localAttributesModified = false;

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
TradeItems AutoStatus/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
logFile.writeln(xmlFilePath);
var xmlFile = getDocByPath(xmlFilePath);

```

```

var xmlDoc = new XmlDocument(xmlFile);

var globalSpecMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
var localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");

    foreachXMLNode("Item/")
    {
        var hmAttributeGlobalPathsToValues = [];
        var hmAttributeLocalPathsToValues = [];
        var bProcessVariant = true;
        if(globalSpecMap != null)
        {
            var hmDstPathsToSrcPaths = [];
            var hmDstPathsToSrcPaths = globalSpecMap.getMappings();

            foreachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
            {
                if(!dstPath.endsWith("ProductType"))
                {

                    if(dstPath.endsWith("GlobalTradeItemNumber"))
                    {
                        sGTIN =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                        logFile.writeln(dstPath+", "+sGTIN);
                    }
                    else
                    {
                        var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                        if(value != "")
                        {

                            hmAttributeGlobalPathsToValues[dstPath] = value;

                            logFile.writeln(dstPath+", "+value);
                                globalAttributesModified =
true;
                                }
                            }
                        }
                    }
                }

                // Now set local attributes

                if(localSpecMap != null)
                {
                    hmDstPathsToSrcPaths = [];
                    hmDstPathsToSrcPaths = localSpecMap.getMappings();

                    foreachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
                    {
                        if(dstPath.endsWith("TargetMarket"))
                        {
                            sTM =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "XX")
;
                            hmAttributeLocalPathsToValues[dstPath] =
sTM;
                            logFile.writeln(dstPath+", "+sTM);

```

```

        }
        else if(dstPath.endsWith("InformationProvider"))
        {
            sip =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "XX")
;
            hmAttributeLocalPathsToValues[dstPath] =
sip;
            logFile.writeln(dstPath+", "+sip);
        }
        else
        {
            var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            if(value != "")
            {
                hmAttributeLocalPathsToValues[dstPath] = value;
                logFile.writeln(dstPath+", "+value);
                localAttributesModified = true;
            }
        }
    }
}

if(sGTIN != "")
{
    tradeItem = getTradeItemSS(sGTIN, sTM, sip);

    if(tradeItem != null)
    {
        // Get the current global item status
        var globalStatus =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Attributes_Spec/GD
SItemStatus"), "");
        logFile.writeln("Global Status is "+globalStatus);
        var localStatus =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_S
pec/GDSItemStatus"), "");
        logFile.writeln("Local Status is "+localStatus);

        if(globalStatus != "" && globalAttributesModified
== true)
        {
            var globalStatusToBeSet =
getToStatus(globalStatus, false);
            if(globalStatusToBeSet != "")
            {
                tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemStatu
s", globalStatusToBeSet);
                logFile.writeln("Setting Global
status to "+globalStatusToBeSet);
            }
            else
            {
                logFile.writeln("Could not retrieve
Global Status from Lookup table");
            }
        }

        if((localAttributesModified == true &&
globalStatus != "") && bProcessVariant == true)

```

```

        {
            logFile.writeln(" localStatus is
:"+localStatus);
            if(localStatus == "")
            {
                // Local item is currently nor
                // Update the global item status to
                Pending Variant Creation.and set the Local Item status to Draft

                tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemStatu
s","Pending Variant Creation");
                var globalItemSaved =
                tradeItem.saveTradeItemSS();
                logFile.writeln("Global Item Saved =
"+globalItemSaved);
                localStatus = "Draft";
                tradeItem =
                getTradeItemSS(sGTIN,sTM,sIP);
            }

            var localStatusToBeSet =
            getToStatus(localStatus,false);
            if(localStatusToBeSet != "")
            {

                tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_Spec/GDSite
mStatus",localStatusToBeSet);
                logFile.writeln("Setting Local status
to "+localStatusToBeSet);
            }
            else
            {
                logFile.writeln("Could not retrieve
Local Status from Lookup table");
            }

            // Setting attributes to make the TradeItem
            compliant

            tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_Spec/EANUCC
/EANUCCType","EN");

            tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_Spec/EANUCC
/EANUCCCode","111111111111");

            tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_Spec/Measur
eCharacteristics/NetWeight/Value","10");

            tradeItem.setTradeItemAttributeSS("Global_Local_Attributes_Spec/Measur
eCharacteristics/NetWeight/UOM","KG");

        }

        if(hmAttributeGlobalPathsToValues != null &&
hmAttributeGlobalPathsToValues.size() != 0)
        {
            logFile.writeln("setting Global attributes
"+hmAttributeGlobalPathsToValues);

            tradeItem.setTradeItemAttributesSS(hmAttributeGlobalPathsToValues);

```

```

    }

    if((hmAttributeLocalPathsToValues != null &&
hmAttributeLocalPathsToValues.size() != 0) && bProcessVariant == true)
    {
        logFile.writeln("setting Local attributes
"+hmAttributeLocalPathsToValues);

        tradeItem.setTradeItemAttributesSS(hmAttributeLocalPathsToValues);
    }

    var tradeItemValidity = [];
//    tradeItemValidity = tradeItem.validateTradeItem();

    if(tradeItemValidity != null &&
tradeItemValidity.size() != 0)
    {
        logFile.writeln("Trade Item not
Valid...Error Attributes are...");

        forEachHmElement(tradeItemValidity,attributePath, errorMsg)
        {
            logFile.writeln(attributePath+" , "
+errorMsg);
        }
    }
    else
    {
        var tradeItemSaved = false;
        catchError(saveError)
        {
            tradeItemSaved =
tradeItem.saveTradeItemSS();
        }
        if(saveError != null || tradeItemSaved ==
false)
        {
            logFile.writeln("There was an
error/failure while saving the tradeitem...");
        }
        else
        {

            logFile.writeln("=====");
            logFile.writeln("Item Saved ");

            logFile.writeln("====="+tradeItemSaved+"=====
=====");

            logFile.writeln(tradeItem);

            if(txn != "" && txn != null)
            {

                logFile.writeln("=====");
                logFile.writeln("Creating
Transaction");
                var bCreateTxn =
createTransaction(tradeItem,txn,"master");
                //
                logFile.writeln((bCreateTxn == true) ? "Done":"Failure");

                logFile.writeln("====="+bCreateTxn+"=====
=====");
            }
        }
    }
}

```

```

    }

    if(notf != "" && notf != null)
    {

        logFile.writeln("=====");
        logFile.writeln("Creating
Notification");
        var bCreateNotf =
createNotification(tradeItem,notf,"master");
        //
        logFile.writeln((bCreateNotf == true) ? "Done":"Failure");
        logFile.writeln("===== "+bCreateNotf+"=====
=====");
    }
}
}
}
}
}
}
}
if(oError != null)
{
    logFile.save("GDS Sample Import - Update Global And Local Items - Auto
Status");
}
logFile.save("GDS Sample Import - Update Global And Local Items - Auto
Status");

```

The following table shows the rules that govern auto status of trade items. In the sample script shown above, a lookup table with the name TradeItem\_AutoStatus\_Lookup is referenced which is the same as shown.

From Status	To Status	Compliance Check Required	Item Type	Notification	Transaction
Draft	Draft	NO	Local Item and Global Item		0
Draft With Variant	Draft With Variant	NO	Global Item		
Global Attributes Modified Locally	Global Attributes Modified Locally	YES	Global Item		0
Global Enrichment Completed	Global Enrichment Completed	NO	Global Item		
Pending Variant Creation	Pending Variant Creation	NO	Global Item		
Modified Compliant	Modified Compliant	YES	Global Item		



From Status	To Status	Compliance Check Required	Item Type	Notification	Transaction
Modified Compliant Approval Pending	Modified Compliant Approval Pending	YES	Global Item		
Modified Locally	Modified Compliant Approval Pending	YES	Global Item	MODIFY_ITEM_APPROVAL_PENDING	MODIFY_ITEM_APPROVAL_PENDING
Complete	Edited	YES	Local Item		0
Compliant	Edited	YES	Local Item		0
Compliant	Compliant	YES	Local Item		
Approved	Edited	NO	Local Item		0
Rejected	Compliant	YES	Local Item		
Registered	Modified Locally	YES	Local Item and Global Item	MODIFY_ITEM_APPROVED	MODIFY_ITEM_APPROVED

### Import Trade Item and Ensure that it is Compliant with a Particular Exchange

Refer the scripts provided for Auto-update the Status of an Item after Import. Also refer the [Handling Validations](#) section.

### Import Trade Item Links in a Particular Status

```

logFile = createOtherOut("GDS Sample Import - Import TradeItem Links");

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import - Import TradeItem Links/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

foreachXMLNode("Links/Link")
{
    var sParentGTIN = checkString(parseXMLNode("Parent/ParentGTIN"), "");
    out.writeln(sParentGTIN);
    if(sParentGTIN != "")
    {
        foreachXMLNode("Child")
        {
            var childGTIN =
checkString(parseXMLNode("ChildGTIN"), "");
            var quantity = checkInt(parseXMLNode("Quantity"), 0);
            var status = checkString(parseXMLNode("Status"), "");
            out.writeln(childGTIN);

            if(childGTIN != "")
            {

```

```

        var tradeItemLink = new
TradeItemLink(sParentGTIN,childGTIN,quantity);
        if(tradeItemLink != null)
        {
            if(status != "")
            {
                tradeItemLink.setLinkAttribute("Status",status);
            }

            var linkSaved =
tradeItemLink.saveTradeItemLink();

            logFile.writeln("=====");
            logFile.writeln("TradeItem Link
"+sParentGTIN+" <==>"+childGTIN+" Saved");

            logFile.writeln("====="+linkSaved+"=====
=====");

            logFile.writeln("=====");
            logFile.writeln("Creating Transaction");
            var bCreateTxn =
createTransaction(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING","master")
;
            // logFile.writeln((bCreateTxn ==
true) ? "Done":"Failure");

            logFile.writeln("====="+bCreateTxn+"=====
=====");

            logFile.writeln("=====");
            logFile.writeln("Creating Notification");
            var bCreateNotf =
createNotification(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING","master"
);
            // logFile.writeln((bCreateNotf
== true) ? "Done":"Failure");

            logFile.writeln("====="+bCreateNotf+"=====
=====");

        }
    }
}

logFile.save("GDS Sample Import - Import TradeItem Links");

```

## Sample XML

The sample XML for trade item link is as shown.

```
<TrigoXML>
<Header>
<Transaction_Id></Transaction_Id>
<File_Id>01</File_Id>
<Record_Id>01</Record_Id>
<Creation_Date>09092004</Creation_Date>
<Creation_Time>161226</Creation_Time>
</Header>
<Links>
<Link>
<Parent>
<ParentGTIN>30000000000001</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>11111111111113</ChildGTIN>
<Quantity>1</Quantity>
<Status>Approved</Status>
</Child>
</Link>
<Link>
<Parent>
<ParentGTIN>30000000000001</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>55555555555555</ChildGTIN>
<Quantity>1</Quantity>
<Status>Approved</Status>
</Child>
</Link>
<Link>
<Parent>
<ParentGTIN>65555555555552</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>55555555555555</ChildGTIN>
<Quantity>1</Quantity>
<Status>Approved</Status>
</Child>
</Link>
<Link>
<Parent>
<ParentGTIN>79819384738924</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>11111111111113</ChildGTIN>
<Quantity>1</Quantity>
<Status>Draft</Status>
</Child>
</Link>
</Links>
</TrigoXML>
```

## Update Trade Item Links through Import

```
function getToStatus(strStatus,checkCompliance)
{
    var setStatus = "";
    var tradeItemLinksStatusSet =
tradeItemStatusLkp.getCtgItemByAttributeValue("TradingItemLink_AutoStatus_Loo
kup_Spec/From_Status",strStatus);
    if(tradeItemLinksStatusSet != null)
    {
        foreachItemSetElement(tradeItemLinksStatusSet, statusItem)
        {
            var toStatus =
checkString(statusItem.getCtgItemAttrib("TradingItemLink_AutoStatus_Lookup_Sp
ec/To_Status"),"");
            if(toStatus != "")
            {
                setStatus = toStatus;
                notif =
checkString(statusItem.getCtgItemAttrib("TradingItemLink_AutoStatus_Lookup_Sp
ec/Notification"),"");
                txn =
checkString(statusItem.getCtgItemAttrib("TradingItemLink_AutoStatus_Lookup_Sp
ec/Transaction"),"");
                break;
            }
        }
    }

    return setStatus;
}

logfile = createOtherOut("GDS Sample Import - Update TradeItem Links");
var tradeItemStatusLkp = getCtgByName("TradingItemLink_AutoStatus_Lookup");

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
Update TradeItem Links/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

foreachXMLNode("Links/Link")
{
    var sParentGTIN = checkString(parseXMLNode("Parent/ParentGTIN"),"");
    out.writeln(sParentGTIN);
    if(sParentGTIN != "")
    {
        foreachXMLNode("Child")
        {
            var childGTIN =
checkString(parseXMLNode("ChildGTIN"),"");
            var quantity = checkInt(parseXMLNode("Quantity"),0);
            var status = checkString(parseXMLNode("Status"),"");
            out.writeln(childGTIN);

            if(childGTIN != "" && sParentGTIN != "")
            {

```

```

        var tradeItemLink = getTradeItemLink(sParentGTIN,
childGTIN);
        if (tradeItemLink != null)
        {

            tradeItemLink.setLinkAttribute("Quantity", "0");

            if (status != "")
            {

                tradeItemLink.setLinkAttribute("Status", status);
            }
            else
            {

                var currentStatus =
tradeItemLink.getLinkAttributeValue("Status");
                var toStatus =
checkString(getToStatus(currentStatus, false), "");
                if (toStatus != "")
                {

                    tradeItemLink.setLinkAttribute("Status", toStatus);
                }
            }

            var linkSaved =
tradeItemLink.saveTradeItemLink();

            logFile.writeln("=====");
            logFile.writeln("TradeItem Link
"+sParentGTIN+" <==>"+childGTIN+" Saved");

            logFile.writeln("====="+linkSaved+"=====
====");

            if (txn != "" && txn != null)
            {

                logFile.writeln("=====");
                logFile.writeln("Creating
Transaction");
                var bCreateTxn =
createTransaction(tradeItemLink, txn, "master");
                //
                logFile.writeln((bCreateTxn == true) ? "Done":"Failure");

                logFile.writeln("====="+bCreateTxn+"=====
====");

            }

            if (notf != "" && notf != null)
            {

                logFile.writeln("=====");
                logFile.writeln("Creating
Notification");
                var bCreateNotf =
createNotification(tradeItemLink, notf, "master");
                //
                logFile.writeln((bCreateNotf == true) ? "Done":"Failure");

```

```

        logFile.writeln("====="+bCreateNotf+"=====");
    }
}

}

}

}

}

logFile.save("GDS Sample Import - Update TradeItem Links");

```

### Auto-update the Status of Trade Item Links after Import

Trade Item Links can be updated with the auto status functionality. It means that if a trade item link is being updated and the status is not set in the xml file, the script can correctly set the status of the trade item link based on its existing status in the system. The following table shows the rules that govern auto status of trade item links. In the sample script shown above, a lookup table with the name TradingItemLink\_AutoStatus\_Lookup is referenced which is the same as shown.

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

```

logFile = createOtherOut("GDS Sample Import - New TradingPartners");

var sGLN = "";
var sTPName = "";
var sCountry = "";
var sGLNID = "";
var status = "";
var tpSaved = false;

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import - New TradingPartners/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

foreachXMLNode("TradingPartner/")
{

```

```

        sGLN = checkString(parseXMLNode("Attribs/GlobalLocationNumber"), "");
        sTPName = checkString(parseXMLNode("Attribs/TradingPartnerName"), "");
        sCountry = checkString(parseXMLNode("Attribs/Country"), "");
        sGLNID = checkString(parseXMLNode("Attribs/GLNIdentifier"), "");
        status = checkString(parseXMLNode("Attribs/Status"), "");

        if(sGLN != "" && sTPName != "" && sCountry != "" && sGLNID != "" &&
status != "")
        {
            var tradingPartner = getTradingPartner(sGLN);

            if(tradingPartner == null)
            {
                tradingPartner = new
TradingPartner(sGLN,sTPName,sCountry,sGLNID);
                if(status != null && status != "")
                {

                    tradingPartner.setPartnerAttribute("GLN_Spec/Status",status);
                }

                tpSaved = tradingPartner.saveTradingPartner();

                logFile.writeln("=====");
                logFile.writeln("Trading Partner Saved");

                logFile.writeln("====="+tpSaved+"=====
==");

                logFile.writeln("=====");
                logFile.writeln("Creating Transaction");
                var bCreateTxn =
createTransaction(tradingPartner,"NEW_TRADING_PARTNER_APPROVED","master");
                //          logFile.writeln((bCreateTxn == true) ?
"Done":"Failure");

                logFile.writeln("====="+bCreateTxn+"=====
=====");

                logFile.writeln("=====");
                logFile.writeln("Creating Notification");
                var bCreateNotf =
createNotification(tradingPartner,"NEW_TRADING_PARTNER_APPROVED","master");
                //          logFile.writeln((bCreateNotf == true) ?
"Done":"Failure");

                logFile.writeln("====="+bCreateNotf+"=====
=====");

                logFile.save("GDS Sample Import - New TradingPartners");
            }
            else
            {
                logFile.writeln("Trading Partner "+sGLN+" already
exists");
            }
        }
        else
        {

```

```

        }
        logFile.writeln("Some Mandatory Attribute is missing");
    }
}

```

## Sample XML

The Sample XML file for TradingPartners is given below

```

<TrigoXML>
<Header>
<Transaction_Id></Transaction_Id>
<File_Id>01</File_Id>
<Record_Id>01</Record_Id>
<Creation_Date>08092004</Creation_Date>
<Creation_Time>114441</Creation_Time>
</Header>
<TradingPartner>
<Attribs>
<GlobalLocationNumber>4444444444772</GlobalLocationNumber>
<TradingPartnerName>test</TradingPartnerName>
<GLNIdentifier>Retailer</GLNIdentifier>
<Country>Egypt</Country>
<Status>Draft</Status>
</Attribs>
</TradingPartner>
<TradingPartner>
<Attribs>
<GlobalLocationNumber>4444576777779</GlobalLocationNumber>
<TradingPartnerName>test ff</TradingPartnerName>
<GLNIdentifier>Manufacturer</GLNIdentifier>
<Country>Egypt</Country>
<Status>Draft</Status>
</Attribs>
</TradingPartner>
</TrigoXML>

```

## Import Trading Partners in a Particular Status

```

function getToStatus(strStatus,checkCompliance)
{
    var setStatus = "";
    var tradingPartnerStatusSet =
tradingPartnerStatusLkp.getCtgItemByAttributeValue("TradingPartner_AutoStatus
_Lookup_Spec/From_Status",strStatus);
    if(tradingPartnerStatusSet != null)
    {
        forEachItemSetElement(tradingPartnerStatusSet, statusItem)
        {
            var toStatus =
checkString(statusItem.getCtgItemAttrib("TradingPartner_AutoStatus_Lookup_Spe
c/To_Status"), "");
            if(toStatus != "")
            {
                setStatus = toStatus;
                notf =
checkString(statusItem.getCtgItemAttrib("TradingPartner_AutoStatus_Lookup_Spe
c/Notification"), "");
                txn =
checkString(statusItem.getCtgItemAttrib("TradingPartner_AutoStatus_Lookup_Spe
c/Transaction"), "");
                break;
            }
        }
    }
}

```



```

        }
    }
    }
    logFile.writeln(setStatus);
    return setStatus;
}

logFile = createOtherOut("GDS Sample Import - Update Trading Partner.txt");
var tradingPartnerStatusLkp =
getCtgByName("TradingPartner_AutoStatus_Lookup");

var sGLN = "";
var notf = "";
var txn = "";

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
Update Trading Partner/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);
var hmAttributePathsToValues = [];
var tpSaved = false;

specMap = getSpecMapByName("GDS XML GLN Attributes Spec Map");
if(specMap != null)
{
    foreachXMLNode("TradingPartner/")
    {
        var hmDstPathsToSrcPaths = [];
        var hmDstPathsToSrcPaths = specMap.getMappings();

        foreachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
        {
            if(srcPath.endsWith("GlobalLocationNumber"))
            {
                sGLN =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            }
            else
            {
                var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                if(value != "")
                {
                    hmAttributePathsToValues[dstPath] = value;
                }
            }
        }
    }

    if(sGLN != null || sGLN != "")
    {
        var tradingPartner = getTradingPartner(sGLN);
        if(tradingPartner != null)
        {
            if(hmAttributePathsToValues != null &&
hmAttributePathsToValues.size() != 0)
            {

```

```

        tradingPartner.setPartnerAttributes(hmAttributePathsToValues);
    }

    var xmlTPStatus =
checkString(hmAttributePathsToValues["GLN_Spec/Status"], "");

    if(xmlTPStatus == "")
    {
        var tpStatus =
checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Status"), "");

        if(tpStatus != "")
        {
            var status =
checkString(getToStatus(tpStatus, false), "");
            if(status != "")
            {

tradingPartner.setPartnerAttribute("GLN_Spec/Status", status);
            }
        }
    }

    tpSaved = tradingPartner.saveTradingPartner();
}

logFile.writeln("=====");
logFile.writeln("Trading Partner Saved");

logFile.writeln("====="+tpSaved+"=====
==");

    if(txn != "" && txn != null)
    {

logFile.writeln("=====");
logFile.writeln("Creating Transaction");
var bCreateTxn =
createTransaction(tradingPartner, txn, "master");
// logFile.writeln((bCreateTxn == true) ?
"Done":"Failure");

logFile.writeln("====="+bCreateTxn+"=====
====");
    }

    if(notf != "" && notf != null)
    {

logFile.writeln("=====");
logFile.writeln("Creating Notification");
var bCreateNotf =
createNotification(tradingPartner, notf, "master");
// logFile.writeln((bCreateNotf == true) ?
"Done":"Failure");

logFile.writeln("====="+bCreateNotf+"=====
====");
    }

```

```

        logFile.save("GDS Sample Import - Update Trading
Partner.txt");
    }
}

```

### Auto-update the Status of the Trading Partner after Import

Trading Partners can be updated with the auto status functionality. It means that if a trading partner is being updated and the status is not set in the xml file, the script can correctly set the status of the trading partner based on its existing status in the system. The following table shows the rules that govern auto status of trading partners. In the sample script shown above a lookup table with the name TradingPartner AutoStatus Lookup is referenced which is the same as shown.

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

### Catalog to Catalog Import

For information about catalog to catalog import, refer the *WebSphere Product Version 5.0 User Guide*.

### Notifications after Import

Most of the scripts above demonstrate how notifications can be created using createNotification() script operation.

### For Export

The various scenarios for export are:

- [Export Trade Items that have changed after some day or newly synched items](#)
- [Export Trade Items which are in a Particular Status](#)
- [Export Trade Items which belong to a Particular Category](#)
- [Export Few Attributes of a Trade Item](#)
- [Export Based on Certain Conditions](#)
- [Export Trade Items Based on Local Status](#)
- [Export All Trade Item Links](#)
- [Export Trade Item Links Based on Status](#)
- [Export All Trading Partners](#)
- [Export Trading Partners – Search Based](#)

### Export Trade Items that have changed after some day or newly synched items

```
var logFile = createOtherOut("GDS - TradeItems Modified.xml");

var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

var txnType = inputs["Transaction_Type"];
var startDateStr = reformatDate(inputs["StartDate"],"dd-MMM-yyyy HH:mm","dd-MMM-yyyy");
var endDateStr = reformatDate(inputs["EndDate"],"dd-MMM-yyyy HH:mm","dd-MMM-yyyy");

var startDate = new Date("dd-MMM-yyyy",startDateStr);
var endDate = new Date("dd-MMM-yyyy",endDateStr);

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

libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

if(tradeItems != null && tradeItems.size() != 0)
{
    for(iVar = 0; iVar < tradeItems.size(); iVar++)
    {
        libScript.getFunctionByName("generateXMLForTradeItem").invoke(tradeItems[iVar],logFile);
    }
}
libScript.getFunctionByName("generateXMLFooter").invoke(logFile);

logFile.save("GDS - TradeItems Modified.xml");
```

### Export Trade Items which are in a Particular Status

```
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

logFile = createOtherOut("GDS - TradeItems.xml");
catchError(oErr)
{
    var hmSearchKeysToValues = [];
    forEachHmElement(inputs,searchKey,searchValue)
    {
        if(checkString(searchValue,"") != "")
        {
            hmSearchKeysToValues[searchKey] = searchValue;
        }
    }

    var tradeItems = [];
    tradeItems = searchTradeItems(hmSearchKeysToValues);

    libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

    if(tradeItems != null && tradeItems.size() != 0)
    {
        for(iVar = 0; iVar < tradeItems.size(); iVar++)
        {
            libScript.getFunctionByName("generateXMLForTradeItem").invoke(tradeItems[iVar],logFile);
        }
    }
}
```

```

    }

    libScript.getFunctionByName("generateXMLFooter").invoke(logFile);
}

logFile.save("GDS - TradeItems.xml");

```

### Export Trade Items which belong to a Particular Category

Refer the script given for Export Items which are in a Particular Status.

### Export Few Attributes of a Trade Item

Refer the script given for Export Items which are in a Particular Status.

The Exports scripts above which search for trade items based on status or category make use of an input spec whose xml is given below. Using this input spec, the same level of search functionality that is provided in the GUI can be achieved.

```

<?xml version="1.0" encoding="UTF-8"?>
<TrigoXML>
  <Header>
    <Type>Spec</Type>
    <Action>Create</Action>
    <CompanyName>supply-side</CompanyName>
    <SpecName>GDS Sample Export - Search Input Spec</SpecName>
    <Attribs>
      <Attrib>
        <Name>TYPE</Name>
        <Value>SCRIPT_INPUT_SPEC</Value>
      </Attrib>
    </Attribs>
  </Header>
  <Nodes>
    <Node>
      <Name>GTIN</Name>
      <Type>ATTRIBUTE</Type>
      <Attribs>
        <Attrib>
          <Name>MAXLENGTH</Name>
          <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
          <Name>MAX_OCCURRENCE</Name>
          <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
          <Name>MIN_OCCURRENCE</Name>
          <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
          <Name>TYPE</Name>
          <Value><![CDATA[STRING]]></Value>
        </Attrib>
      </Attribs>
    </Node>
    <Node>
      <Name>GTIN_NAME</Name>
      <Type>ATTRIBUTE</Type>
      <Attribs>
        <Attrib>
          <Name>MAXLENGTH</Name>
          <Value><![CDATA[3000]]></Value>
        </Attrib>
      </Attribs>
    </Node>
  </Nodes>
</TrigoXML>

```

```

        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>GLOBAL_STATUS</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>LOOKUP_TABLE</Name>
            <Value><![CDATA[Item_Status_LookUp]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>BRAND_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
</Node>

```

```

<Name>PRODUCT_NAME</Name>
<Type>ATTRIBUTE</Type>
<Attribs>
  <Attrib>
    <Name>MAXLENGTH</Name>
    <Value><![CDATA[3000]]></Value>
  </Attrib>
  <Attrib>
    <Name>MAX_OCCURRENCE</Name>
    <Value><![CDATA[1]]></Value>
  </Attrib>
  <Attrib>
    <Name>MIN_OCCURRENCE</Name>
    <Value><![CDATA[0]]></Value>
  </Attrib>
  <Attrib>
    <Name>TYPE</Name>
    <Value><![CDATA[STRING]]></Value>
  </Attrib>
</Attribs>
</Node>
<Node>
  <Name>DESCRIPTION</Name>
  <Type>ATTRIBUTE</Type>
  <Attribs>
    <Attrib>
      <Name>MAXLENGTH</Name>
      <Value><![CDATA[3000]]></Value>
    </Attrib>
    <Attrib>
      <Name>MAX_OCCURRENCE</Name>
      <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
      <Name>MIN_OCCURRENCE</Name>
      <Value><![CDATA[0]]></Value>
    </Attrib>
    <Attrib>
      <Name>TYPE</Name>
      <Value><![CDATA[STRING]]></Value>
    </Attrib>
  </Attribs>
</Node>
<Node>
  <Name>TARGET_MARKET</Name>
  <Type>ATTRIBUTE</Type>
  <Attribs>
    <Attrib>
      <Name>MAXLENGTH</Name>
      <Value><![CDATA[3000]]></Value>
    </Attrib>
    <Attrib>
      <Name>LOOKUP_TABLE</Name>
      <Value><![CDATA[Target_Market_LookUp]]></Value>
    </Attrib>
    <Attrib>
      <Name>MAX_OCCURRENCE</Name>
      <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
      <Name>MIN_OCCURRENCE</Name>
      <Value><![CDATA[0]]></Value>
    </Attrib>
  </Attribs>

```

```

        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>LANGUAGE_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>LOOKUP_TABLE</Name>
            <Value><![CDATA[Language_LookUp]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>IP_GLN</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>IP_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>

```



```

        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
    </Attrib>
    <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
    </Attrib>
</Attribs>
</Node>
<Node>
    <Name>BRAND_OWNER_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>MANUFACTURER_GLN</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>MANUFACTURER_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>

```

```

        <Value><![CDATA[3000]]></Value>
    </Attrib>
    <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
    </Attrib>
    <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
    </Attrib>
</Attribs>
</Node>
<Node>
    <Name>PRODUCT_TYPE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>STRING_ENUMERATION_RULE</Name>
            <Value><![CDATA[var hmProdTypes = []];
var ctg = getCtgByName("Product_Type_LookUp");
var ctgSpecName = ctg.getCatalogSpec().getSpecName();
var itemSet = ctg.getItemSetForCatalog();

forEachItemSetElement(itemSet,item)
{
    var prodTypeDesc = checkString(item.getCtgItemAttrib(ctgSpecName+"/Product
Type Description"), "");
    if(prodTypeDesc != "")
    {
        hmProdTypes.add(prodTypeDesc);
    }
}

res = hmProdTypes;]]></Value>
    </Attrib>
    <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
    </Attrib>
    <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING_ENUMERATION]]></Value>
    </Attrib>
</Attribs>
</Node>
<Node>
    <Name>UDEX_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>

```

```

        <Value><![CDATA[3000]]></Value>
    </Attrib>
    <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
    </Attrib>
    <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
    </Attrib>
</Attribs>
</Node>
<Node>
    <Name>UDEX_CATEGORY_DESC</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
</Nodes>
</TrigoXML>

```

## Export Based on Certain Conditions

Use the script given for Export Items which are in a Particular Status.

### Export Trade Items Based on Local Status

```

var status = checkString(inputs["Status"], "");
var ctg = getCtgByName("Global_Local_Catalog");
logFile = createOtherOut("GDS - TradeItems Based On Local Status.xml");
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");
libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

if(status != "")
{
    var selQuery =
    "[Global_Local_Catalog:Global_Local_Attributes_Spec/GDSItemStatus] =
    '"+status+"'";
    var sel = new AdvancedSelection(ctg, "Test Selection", selQuery);
    var itemSet = sel.getItemSetForSelection();

    if(itemSet != null && itemSet.getItemSetSize() != 0)
    {

```

```

        foreachItemSetElement(itemSet,item)
        {
            var gtin_tm_ip = item.getCtgItemPrimarykey();
            var gtin_tm_ip_tokens = gtin_tm_ip.parseDelim("_");

            var gtin = gtin_tm_ip_tokens[0];
            var tm = gtin_tm_ip_tokens[1];
            var ip = gtin_tm_ip_tokens[2];

            var tradeItem = getTradeItemSS(gtin,tm,ip);

            libScript.getFunctionByName("generateXMLForTradeItem").invoke(tradeItem,logFile);
        }
    }

libScript.getFunctionByName("generateXMLFooter").invoke(logFile);

logFile.save("GDS - TradeItems Based On Local Status.xml");

```

## Export All Trade Item Links

```

// All TradeItemLinks xml file.
logFile = createOtherOut("GDS - All TradeItemLinks.xml");

var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

// Generate the Header
libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

var ctg = getCtgByName("TradeCatalog");

// This is For now.....a search can be put in place if there are too many
items in the catalog
// This contains all gtins that are processed. We can use this to ignore
variants.
var hmGTINs = [];

var itemSet = ctg.getItemSetForCatalog();
logFile.writeln("<Links>");
foreachItemSetElement(itemSet,item)
{
    var gtin_tm_ip = item.getCtgItemPrimarykey();
    var gtin_tm_ip_tokens = gtin_tm_ip.parseDelim("_");

    var gtin = gtin_tm_ip_tokens[0];
    var tm = gtin_tm_ip_tokens[1];
    var ip = gtin_tm_ip_tokens[2];

    var tradeItem = getTradeItemSS(gtin,tm,ip);

    // Get all the child links
    var childLinks = tradeItem.getTradeItemChildLinks();

    if(childLinks != null && childLinks.size() != 0 &&
!hmGTINs.containsKey(gtin))
    {

        libScript.getFunctionByName("generateXMLForTradeItemLinks").invoke(gtin,childLinks,logFile);
    }
}

```

```

        hmGTINs[gtin] = "0";
    }
}
logFile.writeln("</Links>");
// Generate the Footer
libScript.getFunctionByName("generateXMLFooter").invoke(logFile);
logFile.save("GDS - All TradeItemLinks.xml");

```

## Export Trade Item Links Based on Status

```

// All TradeItemLinks xml file.
logFile = createOtherOut("GDS - TradeItemLinks Search Based.xml");

var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

// Generate the Header
libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

var status = checkString(inputs["Status"], "");

var gtinTree = getCategoryTreeByName("GTIN_Hierarchy", true);
var gtinCatSet = gtinTree.getCategorySetByAttributeValue("GTIN_Spec/Child
Item/Status", status);

logFile.writeln("<Links>");
if(gtinCatSet != null && gtinCatSet.getCategorySetSize() != 0)
{
    foreachCategorySetElement(gtinCatSet, gtinCat)
    {
        var parentGTIN = gtinCat.getPrimaryKey();
        var childGTIN = gtinCat.getCategoryAttrib("GTIN_Spec/Child
Item/GlobalTradeItemNumber");

        var tradeItemLink = getTradeItemLink(parentGTIN, childGTIN);
        if(tradeItemLink != null &&
tradeItemLink.getLinkAttributeValue("Status") == status)
        {
            var hmTradeItemLinks = [];
            hmTradeItemLinks.add(tradeItemLink);

            libScript.getFunctionByName("generateXMLForTradeItemLinks").invoke(par
entGTIN, hmTradeItemLinks, logFile);
        }
    }
}
logFile.writeln("</Links>");
// Generate the Footer
libScript.getFunctionByName("generateXMLFooter").invoke(logFile);
logFile.save("GDS - TradeItemLinks Search Based.xml");

```

## Export All Trading Partners

```

var ctg = getCtgByName("Trading_Partner_Catalog");
var ctgSpecName = ctg.getCatalogSpec().getSpecName();
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

logFile = createOtherOut("GDS - All TradingPartners.xml");
libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

foreachCtgItem(item)
{

```

```

    var gln = checkString(item.getCtgItemAttrib(ctgSpecName+"/Global Location
Number (GLN)"), "");
    var tradingPartner = getTradingPartner(gln);
    if(tradingPartner != null)
    {

        libScript.getFunctionByName("generateXMLForTradingPartner").invoke(gln
, tradingPartner, logFile);
    }
}

libScript.getFunctionByName("generateXMLFooter").invoke(logFile);
logFile.save("GDS - All TradingPartners.xml");

```

## Export Trading Partners – Search Based

```

var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

logFile = createOtherOut("GDS - TradingPartners Search Based.xml");
catchError(oErr)
{
    var hmSearchKeysToValues = [];
    forEachHmElement(inputs, searchKey, searchValue)
    {
        if(checkString(searchValue, "") != "")
        {
            hmSearchKeysToValues[searchKey] = searchValue;
        }
    }

    var tradingPartners = [];
    tradingPartners = searchTradingPartners(hmSearchKeysToValues);

    libScript.getFunctionByName("generateXMLHeader").invoke(logFile);

    if(tradingPartners != null && tradingPartners.size() != 0)
    {
        for(iVar = 0; iVar < tradingPartners.size(); iVar++)
        {
            var tradingPartner = tradingPartners[iVar];

            libScript.getFunctionByName("generateXMLForTradingPartner").invoke(nul
l, tradingPartner, logFile);
        }
    }

    libScript.getFunctionByName("generateXMLFooter").invoke(logFile);
}

logFile.save("GDS - TradingPartners Search Based.xml");

```

This script has an input spec whose attributes match the keys of the hashmap that needs to be passed as an argument to the searchTradingPartners() function. This way the search functionality that is provided by the GUI can be stimulated.

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.

```

function generateXMLHeader(xmlFile)
{
    xmlFile.writeln("<TrigoXML>");
    xmlFile.writeln("<Header>");
        xmlFile.writeln("<Transaction_Id></Transaction_Id>");
        xmlFile.writeln("<File_Id>01</File_Id>");
        xmlFile.writeln("<Record_Id>01</Record_Id>");

        xmlFile.writeln("<Creation_Date>" + today().formatDate("ddMMyyyy") + "</Cr
eation_Date>");

        xmlFile.writeln("<Creation_Time>" + today().formatDate("HHmmss") + "</Crea
tion_Time>");
        xmlFile.writeln("</Header>");
}

function generateXMLFooter(xmlFile)
{
    xmlFile.writeln("</TrigoXML>");
}

function generateXMLForTradeItem(tradeItem, xmlFile)
{
    xmlFile.writeln("<Item>");
        xmlFile.writeln("<Attribs>");

            xmlFile.writeln("<GlobalAttribs>");

                xmlFile.writeln("<GlobalTradeItemNumber>" + checkString(tradeItem.getTra
deItemAttributeValueSS("Global_Attributes_Spec/GlobalTradeItemNumber"), "") + "<
/GlobalTradeItemNumber>");

                xmlFile.writeln("<GTINName>" + checkString(tradeItem.getTradeItemAttribu
teValueSS("Global_Attributes_Spec/GTINName/en_US"), "") + "</GTINName>");

                // Format the Product Type
                var sProdType =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Attributes_Spec/Pr
oductType"), "");

                var sProdTypeAbbr =
getCtgByName("Product_Type_LookUp").getCtgItemByPrimaryKey(sProdType).getCtgI
temAttrib("Product_Type_Spec/Product Type Description");

                xmlFile.writeln("<ProductType>" + sProdTypeAbbr + "</ProductType>");

                xmlFile.writeln("<ItemStatus>" + checkString(tradeItem.getTradeItemAttri
buteValueSS("Global_Local_Attributes_Spec/GDSItemStatus"), tradeItem.getTradeI
temAttributeValueSS("Global_Attributes_Spec/GDSItemStatus")) + "</ItemStatus>");
                ;

                xmlFile.writeln("<BrandName>" + checkString(tradeItem.getTradeItemAttrib
uteValueSS("Global_Attributes_Spec/Brand/BrandName"), "") + "</BrandName>");

                xmlFile.writeln("<Pack>" + checkString(tradeItem.getTradeItemAttributeVa
lueSS("Global_Attributes_Spec/Pack"), "") + "</Pack>");

                xmlFile.writeln("<Classification>");

```

```

        xmlFile.writeln("<GlobalClassificationCode>" + tradeItem.getItems()[0].getCtgItemCategories("UDEX_Hierarchy_UCCNet")[0].getPrimaryKey() + "</GlobalClassificationCode>");

        xmlFile.writeln("<GlobalClassificationDescription>" + tradeItem.getItems()[0].getCtgItemCategories("UDEX_Hierarchy_UCCNet")[0].getDisplayValue(getUserLocale()) + "</GlobalClassificationDescription>");

        xmlFile.writeln("<InternalClassificationCode>" + tradeItem.getItems()[0].getCtgItemCategories("Internal_Hierarchy")[0].getPrimaryKey() + "</InternalClassificationCode>");

        xmlFile.writeln("<InternalClassificationDescription>" + tradeItem.getItems()[0].getCtgItemCategories("Internal_Hierarchy")[0].getDisplayValue(getUserLocale()) + "</InternalClassificationDescription>");
        xmlFile.writeln("</Classification>");
        xmlFile.writeln("</GlobalAttribs>");

        xmlFile.writeln("<LocalAttribs>");

        xmlFile.writeln("<TargetMarket>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/TargetMarket"), "") + "</TargetMarket>");

        xmlFile.writeln("<InformationProvider>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/InformationProvider"), "") + "</InformationProvider>");

        xmlFile.writeln("<InformationProviderName>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/InformationProviderName"), "") + "</InformationProviderName>");

        xmlFile.writeln("<ManufacturerGLN>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/ManufacturerGLN"), "") + "</ManufacturerGLN>");

        xmlFile.writeln("<IsPrivate>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/IsPrivate"), "") + "</IsPrivate>");

        xmlFile.writeln("<StartAvailabilityDate>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/DateInformation/StartAvailabilityDate"), "") + "</StartAvailabilityDate>");
        xmlFile.writeln("<MeasurementCharacteristics>");

        xmlFile.writeln("<Height>");

        xmlFile.writeln("<UOM>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/MeasureCharacteristics/Height/UOM"), "") + "</UOM>");

        xmlFile.writeln("<Value>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/MeasureCharacteristics/Height/Value"), "") + "</Value>");
        xmlFile.writeln("</Height>");

        xmlFile.writeln("<Width>");

        xmlFile.writeln("<UOM>" + checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/MeasureCharacteristics/Width/UOM"), "") + "</UOM>");

```



```

        xmlFile.writeln("<Value>" + checkString(tradeItem.getTradeItemAttributeV
alueSS("Global_Local_Attributes_Spec/MeasureCharacteristics/Width/Value"), "")
+ "</Value>");

        xmlFile.writeln("</Width>");

        xmlFile.writeln("<Depth>");

        xmlFile.writeln("<UOM>" + checkString(tradeItem.getTradeItemAttributeVal
ueSS("Global_Local_Attributes_Spec/MeasureCharacteristics/Depth/UOM"), "") + "</
UOM>");

        xmlFile.writeln("<Value>" + checkString(tradeItem.getTradeItemAttributeV
alueSS("Global_Local_Attributes_Spec/MeasureCharacteristics/Depth/Value"), "")
+ "</Value>");

        xmlFile.writeln("</Depth>");

        xmlFile.writeln("</MeasurementCharacteristics>");

        xmlFile.writeln("<OrderingUnitIndicator>" + checkString(tradeItem.getTra
deItemAttributeValueSS("Global_Local_Attributes_Spec/UnitIndicator/OrderingUn
itIndicator"), "") + "</OrderingUnitIndicator>");
        xmlFile.writeln("</LocalAttribs>");

        xmlFile.writeln("</Attribs>");
        xmlFile.writeln("</Item>");
    }

function generateXMLForTradeItemLinks(gtin, tradeItemChildLinks, logFile)
{
    forEachHmElement(tradeItemChildLinks, dummy, tradeItemLink)
    {
        logFile.writeln("<Link>");
        logFile.writeln("<Parent>");

        logFile.writeln("<ParentGTIN>" + gtin + "</ParentGTIN>");
        logFile.writeln("</Parent>");

        logFile.writeln("<Child>");

        logFile.writeln("<ChildGTIN>" + checkString(tradeItemLink.getLinkAttribu
teValue("GlobalTradeItemNumber"), "") + "</ChildGTIN>");

        logFile.writeln("<Quantity>" + checkString(tradeItemLink.getLinkAttribu
eValue("Quantity"), "") + "</Quantity>");

        logFile.writeln("<Status>" + checkString(tradeItemLink.getLinkAttributeV
alue("Status"), "") + "</Status>");
        logFile.writeln("</Child>");
        logFile.writeln("</Link>");
    }
}

function generateXMLForTradingPartner(gln, tradingPartner, logFile)
{
    logFile.writeln("<TradingPartner>");
    logFile.writeln("<Attribs>");

    logFile.writeln("<GlobalLocationNumber>" + checkString(gln, tradingPartne
r.getPartnerAttributeValue("GLN_Spec/Global Location
Number (GLN)")) + "</GlobalLocationNumber>");

```

```

        logFile.writeln("<TradingPartnerName>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner Name"), "") + "</TradingPartnerName>");

        logFile.writeln("<GLNIdentifier>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/GLN Identifier"), "") + "</GLNIdentifier>");

        logFile.writeln("<Country>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner Country"), "") + "</Country>");

        logFile.writeln("<Status>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Status"), "") + "</Status>");
        logFile.writeln("</Attribs>");

        logFile.writeln("</TradingPartner>");
    }

```

## 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.

Module	Script Operation Name	Arguments	Description
Notifications	createNotification	IBusinessObject iObj, String notifType, String userName	<ul style="list-style-type: none"> <li>• Creates a notification object and returns true if the notification was successful.</li> <li>• iObj is the Business Object. The business object can be a trade item or trading partner or and item link.</li> <li>• notifType is the notification type. This can be retrieved from the NotificationSecurityMappingTable lookup table.</li> <li>• 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.</li> </ul>
Transaction	createTransaction	IBusinessObject iObj, String txnType, String userName	<ul style="list-style-type: none"> <li>• Creates a transaction object and returns true if the operation was successful.</li> <li>• iObj is the Business Object. The business object can be a trade item or trading partner or and item link.</li> <li>• txnType is the notification</li> </ul>

Module	Script Operation Name	Arguments	Description
			<p>type. This can be retrieved from the NotificationSecurityMappingTable lookup table.</p> <ul style="list-style-type: none"> <li>• 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.</li> </ul>
Trade Item	newTradeItemSS	None	Creates a new (empty) trade item.
	getTradeItemSS	String sGTIN,String sTM,String sIP	Returns a trade item with the given GTIN, TM and IP values, else returns null.
	TradeItem.setTradeItemAttributeSS	String attributePath,String attributeValue	Sets the attribute value for the attribute path of the trade item.
	TradeItem.setTradeItemAttributesSS	HashMap hmAttributePathsToValues	<p>Sets the trade item with the given hashmap of attribute paths and their corresponding values.</p> <p><b>Note:</b> Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.</p>
	TradeItem.saveTradeItemSS	NONE	Saves the trade item to the database and returns true if the save was successful.
	searchTradeItems	HashMap hmSearchParamsToValues	<p>Searches and retrieves trade Items based on a set of search parameters.</p> <p>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.</p> <ul style="list-style-type: none"> <li>• GTIN</li> <li>• GTIN_NAME</li> <li>• BRAND_NAME</li> <li>• PRODUCT_NAME</li> <li>• DESCRIPTION</li> <li>• TARGET_MARKET</li> <li>• LANGUAGE_CODE <ul style="list-style-type: none"> <li>• IP_GLN</li> <li>• IP_NAME</li> </ul> </li> <li>• BRAND_OWNER_NAME</li> </ul>

Module	Script Operation Name	Arguments	Description
			<ul style="list-style-type: none"> <li>• MANUFACTURER_GLN</li> <li>• MANUFACTURER_NAME <ul style="list-style-type: none"> <li>• PRODUCT_TYPE</li> <li>• UDEX_CODE</li> </ul> </li> <li>• UDEX_CATEGORY_DESC</li> <li>• GLOBAL_ITEM_STATUS</li> <li>• UNIT_INDICATOR_YES(Value should be YES or NO)</li> <li>• UNIT_INDICATOR_NO(Value should be YES or NO)</li> <li>• UNIT_INDICATOR_NONE (Value should be YES or NO)</li> </ul>
	getTradeItemParentLinks	NONE	Searches for parent trade item links for this trade item.
	getTradeItemChildLinks	NONE	Searches for child trade item links for this trade item.
	getModifiedTradeItems	String transactionType, Date startDate, Date endDate	Searches and retrieves trade items modified between start date and end date for a given transaction type.
	TradeItem.getTradeItemAttributeValueSS	String sAttrPath	Returns the attribute value of the TradeItem.
	TradeItem.getItems	NONE	Gets the Global and the local item for this TradeItem.
	TradeItem.setUDEXCategoryCode	String code	Sets the trade item with the given UDEX category code.
	TradeItem.setInternalCategoryCode	String code	Sets the trade item with the given internal category code.
TradingPartner	new\$TradingPartner	String sGLN	Creates a new Trading Partner with the given GLN.
	getTradingPartner	String sGLN	Returns a TradingPartner with the given GLN, else returns null.
	TradingPartner.setPartnerAttribute	String attributePath,String attributeValue	Sets the attribute value for the attribute path of the Trading Partner.
	TradingPartner.setPartnerAttributes	HashMap hmAttributePathsToValues	Sets the Trading Partner with the given hashmap of attribute paths and their corresponding values. <b>Note:</b> Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.

Module	Script Operation Name	Arguments	Description
	TradingPartner.saveTradingPartner	NONE	Saves the trading partner to the database and returns true if the save was successful.
	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. <ul style="list-style-type: none"> <li>• GLN</li> <li>• PARTNER_NAME</li> <li>• STATUS</li> <li>• CONTACT</li> <li>• PARTY_ROLE</li> <li>• CITY_NAME</li> <li>• COUNTRY_CODE</li> <li>• LANGUAGE</li> </ul>
	TradingPartner.getPartnerAttributeValue	String sAttrPath	Returns the attribute value for the given path in the trade item link.
TradeItemLinks	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. <b>Note:</b> 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. <b>Note:</b> 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.

Module	Script Operation Name	Arguments	Description
	TradeItemLinks:getLinkAttributeValue	String sAttrPath	Returns the attribute value for the given path in the Trade Item Link
	getModifiedTradeItemLinks	TBD	TBD
SpecMap	getMappings	NONE	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

#### 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","TESTING","CASE","10000056","000000001.000000002
.000000012");
var itemSaved = tradeItem.saveTradeItemSS();
out.writeln("TradeItem [44444444444442] Saved ? : "+itemSaved);

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

```

var notfCreated =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_PENDING","master");
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/BrandName","Coca Cola");
tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/Pack","8");
tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemStatus","Global Enrichment Completed");

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

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

var notfCreated =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_COMPLETED","master");
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","000000001.000000002.000000012");

if(tradeItem != null)
{
    tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/Brand/BrandName","Coca Cola");
    tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/Pack","8");
    tradeItem.setTradeItemAttributeSS("Global_Attributes_Spec/GDSItemStatus","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.println("TradeItem [45645666666660] Saved ? : "+itemSaved);

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

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

```

### 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/en_US"] =
"COKE";
    hmAttrPathsToValues["Global_Local_Attributes_Spec/InformationProviderName"] = "TEST IP";
    tradeItem.setTradeItemAttributesSS(hmAttrPathsToValues);
    var itemSaved = tradeItem.saveTradeItemSS();
    out.println("TradeItem [45645666666660] Saved ? : "+itemSaved);
}

```

### Getting Attribute Values of a Trade Item

```

var tradeItem = getTradeItemSS("45645666666660","US","7981315112271");
if(tradeItem != null)
{
    var gtinName =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Attributes_Spec/GTINName/en_US"),"");
    var ip =
checkString(tradeItem.getTradeItemAttributeValueSS("Global_Local_Attributes_Spec/InformationProvider"),"");

    out.println("GTINName is "+gtinName);
    out.println("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 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", "master");
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
TradeItemSS("4564566666677","CHILD","EACH","10000056","000000001.000000002.0
0000012");

var itemSaved = tradeItem.saveTradeItemSS();
out.writeln("TradeItem [4564566666677] Saved ? : "+itemSaved);

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

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

var tradeItemLink = new TradeItemLink("4564566666660","4564566666677",1);

var linkSaved = tradeItemLink.saveTradeItemLink();
out.writeln("4564566666660 < == > 4564566666677 [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);

```

## 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_PENDING","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 = getTradeItemSS("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 = 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 6 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. Also provided are sample scripts which show how the scenarios are implemented.

**Note:** IBM Global Data Synchronization for WebSphere Product Center does not have separate import/export module. For release 1.1.1, WebSphere Product Center 5.1.0.3 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.

### Sample XML

This sample xml file has just the mandatory attributes that is required to create a trade item.

```
<TrigoXML>
  <Header>
    <Transaction_Id></Transaction_Id>
    <File_Id>01</File_Id>
    <Record_Id>01</Record_Id>
    <Creation_Date>11092004</Creation_Date>
    <Creation_Time>123020</Creation_Time>
  </Header>

  <Item>
    <Attribs>
      <GlobalAttribs>
        <EANUCCCode> </EANUCCCode>
        <EANUCCCodeType> </EANUCCCodeType>

        <GlobalTradeItemNumber>1111111111113</GlobalTradeItemNumber>
        <GTINName>test</GTINName>
        <CoreExtensionType> <CoreExtensionType>
        <ProductType>EACH</ProductType>
        <ItemStatus>Draft</ItemStatus>
        <BrandName>test</BrandName>
        <Pack>7</Pack>
        <Classification>

        <GlobalClassificationCode>000000001.000000002.000000012</GlobalClassif
icationCode>
        <GlobalClassificationDescription>ALCOHOL
        BASED COCKTAILS/MIXES</GlobalClassificationDescription>

        <InternalClassificationCode>10000056</InternalClassificationCode>
      </GlobalAttribs>
    </Attribs>
  </Item>
</TrigoXML>
```

```

        <InternalClassificationDescription>Cocktails</InternalClassificationDe
scription>

        </Classification>

        </GlobalAttribs>

        <LocalAttribs>
            <TargetMarket></TargetMarket>
            <InformationProvider></InformationProvider>

            <InformationProviderName></InformationProviderName>
        </LocalAttribs>
    </Attribs>
</Item>
</TrigoXML>

```

## Scenarios and Sample Scripts

This section lists down the various customer scenarios for import and export and also provides sample scripts for those scenarios.

Note: For information about how to import or export trade items, links, and trading partners, refer the *WebSphere Product Center User's Guide*.

### For Import

The various scenarios for import are:

- [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](#)
- [Auto-update the Status of a Trade Item after Import](#)
- [Import the Trade Item and Ensure that it is Compliant with a Particular Exchange](#)
- [Import Trade Item Links in a Particular Status](#)
- [Updating Trade Item Links through Import](#)
- [Auto-update the Status of Trade Item Link after Import](#)
- [Import Trading Partners](#)
- [Import Trading Partners in a Particular Status](#)
- [Auto Update the Status of the Trading Partner after Import](#)



- [Catalog to Catalog Import](#)
- [Notifications after 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.

```

    logFile = createOtherOut("GDS Sample Import - Import Global Items
Only.txt");

var sEANUCCCode = "";
var sGTIN = "";
var sGTINName = "";
var sProductType = "";
var sInternalCategoryCode = "";
var sDataPoolCategoryCode = "";
var status = "";

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
Import Global Items Only/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);
var hmAttributePathsToValues = [];

var specMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
if(specMap != null)
{
    var hmDstPathsToSrcPaths = [];
    var hmDstPathsToSrcPaths = specMap.getMappings();

    forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
    {
        if(dstPath.endsWith("eanuccCode"))
        {
            sEANUCCCode =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
            logFile.writeln(dstPath+", "+sEANUCCCode);
        }
        else if(dstPath.endsWith("eanuccType"))
        {
            sEANUCCCodeType =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
            logFile.writeln(dstPath+", "+sEANUCCCodeType);
        }
        else if(dstPath.endsWith("ProductType"))
        {
            sProductType =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());
            logFile.writeln(dstPath+", "+sProductType);
        }
    }
}

```

```

else
{
    var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
    logFile.writeln(dstPath+", "+value);
    if(value != "")
    {
        hmGlobalAttributePathsToValues[dstPath] =
value;
        logFile.writeln(dstPath+", "+value);
    }
}

sExtensionType =
parseXMLNode("Attribs/GlobalAttribs/CoreExtensionType");
logFile.writeln("ExtensionType, "+sExtensionType);
sInternalCategoryCode =
parseXMLNode("Attribs/GlobalAttribs/Classification/InternalClassificationCode
");
logFile.writeln("InternalCategoryCode, "+sInternalCategoryCode);

if(sEANUCCCode != ""
    && sEANUCCCodeType != ""
    && sExtensionType != ""
    && sProductType != ""
    && sInternalCategoryCode != ""
)
{
    tradeItem = new
TradeItemDS(sEANUCCCode, sEANUCCCodeType, sExtensionType, sProductType, sInternal
CategoryCode);

    if(tradeItem != null && (hmGlobalAttributePathsToValues
!= null && hmGlobalAttributePathsToValues.size() != 0))
    {
        logFile.writeln("Setting attributes for Global
Item "+hmGlobalAttributePathsToValues);

        tradeItem.setTradeItemAttributesDS(hmGlobalAttributePathsToValues);
    }

    tradeItem.saveTradeItemDS();

    createTransaction(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "master
");

    createNotification(tradeItem, "ITEM_GLOBAL_ENRICHMENT_COMPLETED", "maste
r");

    out.writeln("=====");
    out.writeln("Global Item Saved");

    out.writeln("=====");
    out.writeln(tradeItem);

    out.writeln("=====");
    out.writeln("Creating Transaction");

```

```

        var bCreateTxn =
createTransaction(tradeItem,"ITEM_GLOBAL_ENRICHMENT_COMPLETED","master");

        out.writeln("=====");

        out.writeln("=====");
        out.writeln("Creating Notification");
        var bCreateNotf =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_COMPLETED","master");

        out.writeln("=====");
    }
}

logFile.save("GDS Sample Import - Import Global Items Only.txt");

```

## Import Global and Local Trade Items

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// This script inserts a trade item (a global item and its variant) into GDS
// in any given status
//
// Assumptions: -
// *****
// 1. Both Global Item and Local Item are being imported.
// 2. The GDS Item Status is mentioned in the Sample XML file.
// 3. The Notification Type is set to Approved while creating Transaction and
// Notification. This can be set to anything.
// But it should correspond to the Item status given in the XML file. In
// this sample scenario,
// we have set the local item status to Approved and therefore we are
// setting the Notification Type to NEW_ITEM_APPROVED.
// 4. The XML file is taken from the import and it is always the latest file.
// The name of the Import is hardcoded here.
// 5. There are two spec maps created in the system, called,
// a) GDS XML Global Attributes Spec Map (this maps the Mandatory Global
// Attributes to the Sample XML file ).
// b) GDS XML Local Attributes Spec Map (this maps the Mandatory Local
// Attributes to the Sample XML file ).
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function removeTradeItem(tradeItem, sGTIN)
{
    logFile.writeln("Performing Cleanup.....");
    var items = [];
    items = tradeItem.getItems();

    if(items != null && items.size() != 0)
    {
        var globalItem = items[0];
        var localItem = items[1];
        var gtin_tm_ip = "";

        if(localItem != null)
        {

```

```

        logFile.writeln("Local Item Deleted :
"+tradeItem.deleteTMIPVariant());
    }

    if(globalItem != null)
    {
        logFile.writeln("Global and TradeItem Deleted :
"+tradeItem.deleteGTIN());
    }
}

logFile.writeln("Cleanup complete....");
}

var logFile = createOtherOut("GDS Sample TradeItem Import - New
TradeItems.txt");
var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample TradeItem
Import - New TradeItems/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

var globalSpecMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
var localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");
catchError(oErr)
{
    foreachXMLNode("Item/")
    {
        var statusFromXML = "";
        var bValueSetForLocal = false;
        var hmGlobalAttributePathsToValues = [];
        var hmLocalAttributePathsToValues = [];

        if(globalSpecMap != null)
        {
            var sEANUCCCode = "";
            var sEANUCCCodeType = "";
            var sExtensionType = "";
            var sProductType = "";
            var sInternalCategoryCode = "";

            var hmDstPathsToSrcPaths = [];
            var hmDstPathsToSrcPaths = globalSpecMap.getMappings();

            foreachHmElement(hmDstPathsToSrcPaths, dstPath,
srcPath)
            {
                if(dstPath.endsWith("eanuccCode"))
                {
                    sEANUCCCode =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());

                    logFile.writeln(dstPath+", "+sEANUCCCode);
                }
                else if(dstPath.endsWith("eanuccType"))
                {
                    sEANUCCCodeType =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());

```

```

        logFile.writeln(dstPath+", "+sEANUCCCodeType);
    }
    else if(dstPath.endsWith("ProductType"))
    {
        sProductType =
parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName());

        logFile.writeln(dstPath+", "+sProductType);
    }
    else if(dstPath.endsWith("GDSItemStatus"))
    {
        var globalStatusCtg =
getCtgByName("Item_Status_LookUp");
        statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
        if(statusFromXML != "" &&
globalStatusCtg.containsByPrimarykey(statusFromXML))
        {
            hmGlobalAttributePathsToValues[dstPath] = statusFromXML;
        }
        else
        {
            hmGlobalAttributePathsToValues[dstPath] = "Draft";
        }
    }
    else
    {
        var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
        logFile.writeln(dstPath+", "+value);
        if(value != "")
        {
            hmGlobalAttributePathsToValues[dstPath] = value;

            logFile.writeln(dstPath+", "+value);
        }
    }
}

sExtensionType =
parseXMLNode("Attribs/GlobalAttribs/CoreExtensionType");
logFile.writeln("ExtensionType, "+sExtensionType);
sInternalCategoryCode =
parseXMLNode("Attribs/GlobalAttribs/Classification/InternalClassificationCode
");

logFile.writeln("InternalCategoryCode, "+sInternalCategoryCode);

if(sEANUCCCode != ""
    && sEANUCCCodeType != ""
    && sExtensionType != ""
    && sProductType != ""
    && sInternalCategoryCode != ""
)
{

```

```

        tradeItem = new
TradeItemDS (sEANUCCCode,sEANUCCCodeType,sExtensionType,sProductType,sInternal
CategoryCode);

        if(tradeItem != null &&
(hmGlobalAttributePathsToValues != null &&
hmGlobalAttributePathsToValues.size() != 0))
        {
            logFile.writeln("Setting attributes for
Global Item "+hmGlobalAttributePathsToValues);

            tradeItem.setTradeItemAttributesDS(hmGlobalAttributePathsToValues);
        }
    }

    // Now set the Local Item

    if(localSpecMap != null)
    {
        hmDstPathsToSrcPaths = [];
        hmDstPathsToSrcPaths = localSpecMap.getMappings();

        forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
        {
            if(dstPath.endsWith("GDSItemStatus"))
            {
                var localStatusCtg =
getCtgByName("Item_Status_Local_LookUp");
                statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

                if(statusFromXML != "" &&
localStatusCtg.containsByPrimaryKey(statusFromXML))
                {
                    hmLocalAttributePathsToValues[dstPath] = statusFromXML;
                }
                else
                {
                    hmLocalAttributePathsToValues[dstPath] = "Draft";
                }
            }
            else
            {
                var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                if(value != "")
                {
                    hmLocalAttributePathsToValues[dstPath] = value;
                    logFile.writeln(dstPath+", "+value);

                    if(!dstPath.contains("GlobalTradeItemNumber"))
                    {
                        bValueSetForLocal = true;
                    }
                }
            }
        }
    }
}

```

```

        if(bValueSetForLocal == true)
        {
            if(tradeItem != null &&
(hmLocalAttributePathsToValues != null &&
hmLocalAttributePathsToValues.size() != 0))
            {
                logFile.writeln("Setting attributes for
Local Item "+hmLocalAttributePathsToValues);

                tradeItem.setTradeItemAttributesDS(hmLocalAttributePathsToValues);

                if(checkString(hmGlobalAttributePathsToValues["Global_Attributes_Spec/
GDSItemStatus"], "") == "Draft")
                {

                    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemStatu
s", "Draft With Variant");

                    logFile.writeln("Updating Global
status to Draft with Variant since local Item is being inserted");
                }
            }
        }

        var tradeItemSaved = false;

        if(tradeItem != null)
        {
            catchError(saveError)
            {
                tradeItemSaved = tradeItem.saveTradeItemDS();
            }
            if(saveError != null || tradeItemSaved == false)
            {
                logFile.writeln("There was an error/failure while
saving the tradeitem....");
                logFile.writeln(""+saveError);
                logFile.writeln("Now rolling back the
changes....");

                removeTradeItem(tradeItem, tradeItem.getTradeItemAttributeValueDS("Glob
al_Attributes_Spec/GlobalTradeItemNumber"));
            }
            else
            {

                logFile.writeln("=====");
                logFile.writeln("Trade Item Saved");

                logFile.writeln("====="+tradeItemSaved+"=====");

                logFile.writeln(tradeItem);

                logFile.writeln("=====");
                logFile.writeln("Creating Transaction");
                var bCreateTxn =
createTransaction(tradeItem, "NEW_ITEM_APPROVAL_PENDING", "master");

                logFile.writeln("====="+bCreateTxn+"=====");
            }
        }

```

```

        logFile.writeln("=====");
        logFile.writeln("Creating Notification");
        var bCreateNotf =
createNotification(tradeItem,"NEW_ITEM_APPROVAL_PENDING","master");

        logFile.writeln("===== "+bCreateNotf+"=====
=====");
    }
    }
    else
    {
        logFile.writeln("TradeItem is null");
    }
}

logFile.save("GDS Sample TradeItem Import - New TradeItems.txt");
}
if(oErr != null)
{
    logFile.writeln("ERROR!! "+oErr);
    logFile.save("GDS Sample TradeItem Import - New TradeItems.txt");
}

```

### Import Trade Item in a Particular Status

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
// This script updates a trade item into GDS in any given status
//
// Assumptions: -
// *****
// 1. The GDS Item Status is mentioned in the Sample XML file.
// 2. The Notification Type is set to Approved while creating Transaction and
Notification. This can be set to anything.
//    But it should correspond to the Item status given in the XML file.
// 3. The XML file is taken from the import and it is always the latest file.
The name of the Import is hardcoded here.
// 4. There are two spec maps created in the system, called,
//    a) GDS XML Global Attributes Spec Map (this maps the Mandatory Global
Attributes to the Sample XML file ).
//    b) GDS XML Local Attributes Spec Map (this maps the Mandatory Local
Attributes to the Sample XML file ).
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

logFile = createOtherOut("GDS Sample TradeItem Import - Update
TradeItems.txt");
var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample TradeItem
Import - Update TradeItems/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

var globalSpecMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
var localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");

```



```

forEachXMLNode("Item/")
{
    var sEANUCCCode = "";
    var sGTIN = "";
    var sTM = "";
    var sIP = "";
    var localStatusFound = false;

    var hmGlobalAttributePathsToValues = [];
    var hmLocalAttributePathsToValues = [];
    var bCheckGlobalStatus = false;

    if(globalSpecMap != null)
    {
        var hmDstPathsToSrcPaths = [];
        var hmDstPathsToSrcPaths = globalSpecMap.getMappings();

        forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
        {
            if(!dstPath.endsWith("ProductType"))
            {
                if(dstPath.endsWith("eanuccCode"))
                {
                    sEANUCCCode =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                    logFile.writeln(dstPath+", "+sEANUCCCode);
                }
                else if(dstPath.endsWith("GlobalTradeItemNumber"))
                {
                    sGTIN =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                    logFile.writeln(dstPath+", "+sGTIN);
                }
                else if(dstPath.endsWith("GDSItemStatus"))
                {
                    var globalStatusCtg =
getCtgByName("Item_Status_LookUp");
                    statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

                    if(statusFromXML != "" &&
globalStatusCtg.containsByPrimaryKey(statusFromXML))
                    {
                        hmGlobalAttributePathsToValues[dstPath] = statusFromXML;
                    }
                }
                else
                {
                    var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                    logFile.writeln(dstPath+", "+value);
                    if(value != "")
                    {
                        hmGlobalAttributePathsToValues[dstPath] = value;
                        logFile.writeln(dstPath+", "+value);
                    }
                }
            }
        }
    }
}

```

```

// Now set local attributes

localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");

if(localSpecMap != null)
{
    hmDstPathsToSrcPaths = [];
    hmDstPathsToSrcPaths = localSpecMap.getMappings();

    forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
    {
        if(dstPath.endsWith("TargetMarket"))
        {
            sTM =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            hmLocalAttributePathsToValues[dstPath] = sTM;
            logFile.println(dstPath+" "+sTM);
        }
        else if(dstPath.endsWith("InformationProvider"))
        {
            sIP =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            hmLocalAttributePathsToValues[dstPath] = sIP;
            logFile.println(dstPath+" "+sIP);
        }
        else if(dstPath.endsWith("GDSItemStatus"))
        {
            var localStatusCtg =
getCtgByName("Item_Status_Local_LookUp");
            statusFromXML =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

            if(statusFromXML != "" &&
localStatusCtg.containsByPrimaryKey(statusFromXML))
            {
                hmLocalAttributePathsToValues[dstPath] =
statusFromXML;

                bCheckGlobalStatus = true;
            }
        }
        else
        {
            var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
            if(value != "")
            {
                hmLocalAttributePathsToValues[dstPath] =
value;

                logFile.println(dstPath+" "+value);
            }
        }
    }
}

var tradeItem = getTradeItemDS(sEANUCCCode, sTM, sIP, sGTIN);
if(tradeItem != null)
{
    if(bCheckGlobalStatus == true)
    {
        var iItems = tradeItem.getItems();
    }
}

```

```

        if(iItems != null && iItems.size() != 0)
        {
            if(iItems[0] != null && iItems[1] == null) // That
means the local item doesnt exist and we are trying to create one.
            {

                if(checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attribut
es_Spec/GDSItemStatus"), "") == "Draft")
                {

                    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemStatu
s", "Draft With Variant");

                    tradeItem.saveTradeItemDS();
                    logFile.writeln("Updating Global
status to Draft with Variant since local Item is being set");
                }
            }
        }

        if(hmGlobalAttributePathsToValues != null &&
hmGlobalAttributePathsToValues.size() != 0)
        {
            logFile.writeln("setting global attributes
"+hmGlobalAttributePathsToValues);

            tradeItem.setTradeItemAttributesDS(hmGlobalAttributePathsToValues);
        }

        if(hmLocalAttributePathsToValues != null &&
hmLocalAttributePathsToValues.size() != 0)
        {
            logFile.writeln("setting local attributes
"+hmLocalAttributePathsToValues);

            tradeItem.setTradeItemAttributesDS(hmLocalAttributePathsToValues);
        }

        var tradeItemSaved = false;
        catchError(saveError)
        {
            tradeItemSaved = tradeItem.saveTradeItemDS();
        }
        if(saveError != null || tradeItemSaved == false)
        {
            logFile.writeln("There was an error/failure while saving
the tradeitem....");
        }
        else
        {
            logFile.writeln("=====");
            logFile.writeln("Item Saved");

            logFile.writeln("===== "+tradeItemSaved+"=====
");
            logFile.writeln(tradeItem);

            logFile.writeln("=====");
            logFile.writeln("Creating Transaction");
            var bCreateTxn =
createTransaction(tradeItem, "MODIFY_ITEM_APPROVED", "master");

```

```

        logFile.writeln("====="+bCreateTxn+"=====");
    }

    logFile.writeln("=====");
    logFile.writeln("Creating Notification");
    var bCreateNotf =
createNotification(tradeItem,"MODIFY_ITEM_APPROVED","master");

    logFile.writeln("====="+bCreateNotf+"=====");
}
}

logFile.save("GDS Sample TradeItem Import - Update TradeItems.txt");

```

### Update an Already Existing Trade Item through Import

Refer the scripts for Import an item in a particular status and Import an item through auto status

### Auto-update the Status of the Trade Item after Import

Trade items can be updated with the auto status functionality. It means that if a trade item is being updated and the status is not set in the xml file, the script can correctly set the status of the trade item based on its existing status in the system.

```

////////////////////////////////////
////////////////////////////////////
// This script updates a trade item (a global item and its variant) into GDS
// in any given status
//
// Assumptions: -
// *****
// 1. Both Global Item and Local Item are being imported.
// 2. The GDS Item Status is mentioned in the Sample XML file and it is Local
// Item status.
// 3. The Notification Type is set to Approved while creating Transaction and
// Notification. This can be set to anything.
// But it should correspond to the Item status given in the XML file. In
// this sample scenario,
// we have set the local item status to Approved and therefore we are
// setting the Notification Type to NEW_ITEM_APPROVED.
// 4. The XML file is taken from the import and it is always the latest file.
// The name of the Import is hardcoded here.
// 5. There are two spec maps created in the system, called,
// a) GDS XML Global Attributes Spec Map (this maps the Mandatory Global
// Attributes to the Sample XML file ).
// b) GDS XML Local Attributes Spec Map (this maps the Mandatory Local
// Attributes to the Sample XML file ).
////////////////////////////////////
////////////////////////////////////

function getToStatus(strStatus,checkCompliance)
{

```

```

        var setStatus = "";
        var tradeItemStatusSet =
tradeItemStatusLkp.getCtgItemByAttributeValue("TradeItem_AutoStatus_Lookup
Spec/From_Status",strStatus);
        if(tradeItemStatusSet != null)
        {
            foreachItemSetElement(tradeItemStatusSet, statusItem)
            {
                var toStatus =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_Lookup
Spec/To_Status"), "");
                if(toStatus != "")
                {
                    setStatus = toStatus;
                    notf =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_Lookup
Spec/Notification"), "");
                    txn =
checkString(statusItem.getCtgItemAttrib("TradeItem_AutoStatus_Lookup
Spec/Transaction"), "");
                    break;
                }
            }
        }

        return setStatus;
    }

catchError(oError)
{
    logFile = createOtherOut("GDS Sample Import - Update Global And Local Items -
Auto Status");
    var tradeItemStatusLkp = getCtgByName("TradeItem_AutoStatus_Lookup");
    var notf = "";
    var txn = "";

    var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample TradeItem
Import - Update TradeItems With AutoStatus/");
    docFiles = docFiles.sort();

    var xmlFilePath = docFiles[docFiles.size() - 1];
    logFile.writeln(xmlFilePath);
    var xmlFile = getDocByPath(xmlFilePath);
    var xmlDoc = new XmlDocument(xmlFile);

    var globalSpecMap = getSpecMapByName("GDS XML Global Attributes Spec Map");
    var localSpecMap = getSpecMapByName("GDS XML Local Attributes Spec Map");

    foreachXMLNode("Item/")
    {
        var sEANUCCCode = "";
        var sGTIN = "";
        var sTM = "";
        var sIP = "";
        var globalAttributesModified = false;
        var localAttributesModified = false;

        var hmAttributeGlobalPathsToValues = [];
        var hmAttributeLocalPathsToValues = [];
        var bProcessVariant = false;

```

```

        if(globalSpecMap != null)
        {
            var hmDstPathsToSrcPaths = [];
            var hmDstPathsToSrcPaths = globalSpecMap.getMappings();

            forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
            {
                if(!dstPath.endsWith("ProductType"))
                {
                    if(dstPath.endsWith("eanuccCode"))
                    {
                        sEANUCCCode =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");

                        logFile.writeln(dstPath+", "+sEANUCCCode);
                    }
                    else
                    if(dstPath.endsWith("GlobalTradeItemNumber"))
                    {
                        sGTIN =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                        logFile.writeln(dstPath+", "+sGTIN);
                    }
                    else
                    {
                        var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                        if(value != "")
                        {

                            hmAttributeGlobalPathsToValues[dstPath] = value;

                            logFile.writeln(dstPath+", "+value);
                                globalAttributesModified =
true;
                                }
                            }
                        }
                    }
                }

            // Now set local attributes

            if(localSpecMap != null)
            {
                hmDstPathsToSrcPaths = [];
                hmDstPathsToSrcPaths = localSpecMap.getMappings();

                forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
                {
                    if(dstPath.endsWith("TargetMarket"))
                    {
                        sTM =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
                        if(sTM != "")
                        {

                            hmAttributeLocalPathsToValues[dstPath] = sTM;
                                logFile.writeln(dstPath+", "+sTM);
                                localAttributesModified = true;
                                bProcessVariant = true;
                                }
                            }
                        }
                    }
                }
            }
        }
    }

```

```

else if(dstPath.endsWith("InformationProvider"))
{
    sIP =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
    if(sIP != "")
    {
        hmAttributeLocalPathsToValues[dstPath] = sIP;
        logFile.writeln(dstPath+", "+sIP);
        localAttributesModified = true;
        bProcessVariant = true;
    }
}
else
{
    var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
    if(value != "")
    {
        hmAttributeLocalPathsToValues[dstPath] = value;
        logFile.writeln(dstPath+", "+value);
    }
}
}

if(sEANUCCCode != "")
{
    tradeItem = getTradeItemDS(sEANUCCCode, sTM, sIP, sGTIN);

    if(tradeItem != null)
    {
        // Get the current global item status
        var globalStatus =
checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attributes_Spec/GD
SItemStatus"), "");
        logFile.writeln("Global Status is "+globalStatus);
        var localStatus =
checkString(tradeItem.getTradeItemAttributeValueDS("Global_Local_Attributes_S
pec/GDSItemStatus"), "");
        logFile.writeln("Local Status is "+localStatus);

        if(globalStatus != "" && globalAttributesModified
== true)
        {
            var globalStatusToBeSet =
getToStatus(globalStatus, false);
            if(globalStatusToBeSet != "")
            {
                tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemStatu
s", globalStatusToBeSet);
                logFile.writeln("Setting Global
status to "+globalStatusToBeSet);
            }
            else
            {
                logFile.writeln("Could not retrieve
Global Status from Lookup table");
            }
        }
    }
}

```

```

        if((localAttributesModified == true &&
globalStatus != "") && bProcessVariant == true)
        {
            logFile.writeln(" localStatus is
:"+localStatus);
            if(localStatus == "")
            {
                // Local item is currently not
                // Update the global item status to
                Pending Variant Creation.and set the Local Item status to Draft

                tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemStatu
s","Pending Variant Creation");
                var globalItemSaved =
tradeItem.saveTradeItemDS();
                logFile.writeln("Global Item Saved =
"+globalItemSaved);
                localStatus = "Draft";
                tradeItem =
getTradeItemDS(sEANUCCCode,sTM,sIP,sGTIN);
            }

            var localStatusToBeSet =
getToStatus(localStatus,false);
            if(localStatusToBeSet != "")
            {
                tradeItem.setTradeItemAttributeDS("Global_Local_Attributes_Spec/GDSite
mStatus",localStatusToBeSet);
                logFile.writeln("Setting Local status
to "+localStatusToBeSet);
            }
            else
            {
                logFile.writeln("Could not retrieve
Local Status from Lookup table");
            }
        }

        if(hmAttributeGlobalPathsToValues != null &&
hmAttributeGlobalPathsToValues.size() != 0)
        {
            logFile.writeln("setting Global attributes
"+hmAttributeGlobalPathsToValues);

            tradeItem.setTradeItemAttributesDS(hmAttributeGlobalPathsToValues);
        }

        if((hmAttributeLocalPathsToValues != null &&
hmAttributeLocalPathsToValues.size() != 0) && bProcessVariant == true)
        {
            logFile.writeln("setting Local attributes
"+hmAttributeLocalPathsToValues);

            tradeItem.setTradeItemAttributesDS(hmAttributeLocalPathsToValues);
        }

        var tradeItemSaved = false;
        catchError(saveError)
        {

```



```

tradeItemSaved =
tradeItem.saveTradeItemDS();
    }
    if(saveError != null || tradeItemSaved == false)
    {
        logFile.writeln("There was an error/failure
while saving the tradeitem...");
    }
    else
    {

        logFile.writeln("=====");
        logFile.writeln("Item Saved ");

        logFile.writeln("====="+tradeItemSaved+"=====
=====");

        logFile.writeln(tradeItem);

        if(txn != "" && txn != null)
        {

            logFile.writeln("=====");
            logFile.writeln("Creating
Transaction");
            var bCreateTxn =
createTransaction(tradeItem,txn,"master");

            logFile.writeln("====="+bCreateTxn+"=====
=====");

            }

            if(notf != "" && notf != null)
            {

                logFile.writeln("=====");
                logFile.writeln("Creating
Notification");
                var bCreateNotf =
createNotification(tradeItem,notf,"master");

                logFile.writeln("====="+bCreateNotf+"=====
=====");

                }

            }

        }

    }
}
if(oError != null)
{
    logFile.save("GDS Sample Import - Update Global And Local Items - Auto
Status");
}
logFile.save("GDS Sample Import - Update Global And Local Items - Auto
Status");

```

The following table shows the rules that govern auto status of trade items. In the sample script shown above, a lookup table with the name TradeItem\_AutoStatus\_Lookup is referenced which is the same as shown.

From Status	To Status	Item Type	Notification	Transaction
Draft	Draft	Local Item and Global Item		0
Draft With Variant	Draft With Variant	Global Item		
Global Attributes Modified Locally	Global Attributes Modified Locally	Global Item		0
Global Enrichment Completed	Global Enrichment Completed	Global Item		
Pending Variant Creation	Pending Variant Creation	Global Item		
Modified Compliant	Modified Compliant	Global Item		
Modified Compliant Approval Pending	Modified Compliant Approval Pending	Global Item		
Modified Locally	Modified Compliant Approval Pending	Global Item	MODIFY_ITEM_APPROVAL_PENDING	MODIFY_ITEM_APPROVAL_PENDING
Complete	Edited	Local Item		0
Compliant	Edited	Local Item		0
Compliant	Compliant	Local Item		
Approved	Edited	Local Item		0
Rejected	Compliant	Local Item		
Synchronized	Modified Locally	Local Item and Global Item	MODIFY_ITEM_APPROVED	MODIFY_ITEM_APPROVED

### **Import Trade Item and Ensure that it is Compliant with a Particular Exchange**

To check for trade item compliancy a script operation named validateTradeItem() is exposed which can be used within the import scripts to check for tradeitem compliancy with a particular exchange.

## Import Trade Item Links in a Particular Status

```
logFile = createOtherOut("GDS Sample TradeItemLinks Import - All
TradeItemLinks");

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample
TradeItemLinks Import - All TradeItemLinks/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

foreachXMLNode("Links/Link")
{
    var sParentGTIN = checkString(parseXMLNode("Parent/ParentGTIN"), "");
    out.writeln(sParentGTIN);
    if(sParentGTIN != "")
    {
        foreachXMLNode("Child")
        {
            var childGTIN =
checkString(parseXMLNode("ChildGTIN"), "");
            var quantity = checkInt(parseXMLNode("Quantity"), 0);
            var status = checkString(parseXMLNode("Status"), "");
            out.writeln(childGTIN);

            if(childGTIN != "")
            {
                var tradeItemLink = new
TradeItemLink(sParentGTIN, childGTIN, quantity);
                if(tradeItemLink != null)
                {
                    if(status != "")
                    {
                        tradeItemLink.setLinkAttribute("Status", status);
                    }

                    var linkSaved =
tradeItemLink.saveTradeItemLink();

                    logFile.writeln("=====");
                    logFile.writeln("TradeItem Link
"+sParentGTIN+" <==>"+childGTIN+" Saved");
                    logFile.writeln("====="+linkSaved+"=====
====");

                    logFile.writeln("=====");
                    logFile.writeln("Creating Transaction");
                    var bCreateTxn =
createTransaction(tradeItemLink, "MODIFY_ITEM_LINK_APPROVAL_PENDING", "master")
;
                    // logFile.writeln((bCreateTxn ==
true) ? "Done":"Failure");
```

```

        logFile.writeln("====="+bCreateTxn+"=====");

        logFile.writeln("=====");
        logFile.writeln("Creating Notification");
        var bCreateNotf =
createNotification(tradeItemLink,"MODIFY_ITEM_LINK_APPROVAL_PENDING","master"
);
        //          logFile.writeln((bCreateNotf
== true) ? "Done":"Failure");

        logFile.writeln("====="+bCreateNotf+"=====");

    }

}

}

logFile.save("GDS Sample TradeItemLinks Import - All TradeItemLinks");

```

## Sample XML

The sample XML for trade item link is as shown.

```

<TrigoXML>
<Header>
<Transaction_Id></Transaction_Id>
<File_Id>01</File_Id>
<Record_Id>01</Record_Id>
<Creation_Date>09092004</Creation_Date>
<Creation_Time>161226</Creation_Time>
</Header>
<Links>
<Link>
<Parent>
<ParentGTIN>30000000000001</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>11111111111113</ChildGTIN>
<Quantity>1</Quantity>
<Status>Approved</Status>
</Child>
</Link>
<Link>
<Parent>
<ParentGTIN>30000000000001</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>55555555555555</ChildGTIN>
<Quantity>1</Quantity>
<Status>Approved</Status>
</Child>
</Link>
<Link>
<Parent>
<ParentGTIN>65555555555552</ParentGTIN>
</Parent>
<Child>

```

```

<ChildGTIN>5555555555555555</ChildGTIN>
<Quantity>1</Quantity>
<Status>Approved</Status>
</Child>
</Link>
<Link>
<Parent>
<ParentGTIN>79819384738924</ParentGTIN>
</Parent>
<Child>
<ChildGTIN>111111111111113</ChildGTIN>
<Quantity>1</Quantity>
<Status>Draft</Status>
</Child>
</Link>
</Links>
</TrigoXML>

```

## Update Trade Item Links through Import

```

function getToStatus(strStatus,checkCompliance)
{
    var setStatus = "";
    var tradeItemLinksStatusSet =
tradeItemStatusLkp.getCtgItemByAttributeValue("TradingItemLink_AutoStatus_Loo
kup_Spec/From_Status",strStatus);
    if(tradeItemLinksStatusSet != null)
    {
        forEachItemSetElement(tradeItemLinksStatusSet, statusItem)
        {
            var toStatus =
checkString(statusItem.getCtgItemAttrib("TradingItemLink_AutoStatus_Lookup_Sp
ec/To_Status"),"");
            if(toStatus != "")
            {
                setStatus = toStatus;
                notif =
checkString(statusItem.getCtgItemAttrib("TradingItemLink_AutoStatus_Lookup_Sp
ec/Notification"),"");
                txn =
checkString(statusItem.getCtgItemAttrib("TradingItemLink_AutoStatus_Lookup_Sp
ec/Transaction"),"");
                break;
            }
        }
    }

    return setStatus;
}

logFile = createOtherOut("GDS Sample Import - Update TradeItem Links");
var tradeItemStatusLkp = getCtgByName("TradingItemLink_AutoStatus_Lookup");

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample Import -
Update TradeItem Links/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

```

```

foreachXMLNode("Links/Link")
{
    var sParentGTIN = checkString(parseXMLNode("Parent/ParentGTIN"), "");
    out.writeln(sParentGTIN);
    if(sParentGTIN != "")
    {
        foreachXMLNode("Child")
        {
            var childGTIN =
checkString(parseXMLNode("ChildGTIN"), "");
            var quantity = checkInt(parseXMLNode("Quantity"), 0);
            var status = checkString(parseXMLNode("Status"), "");
            out.writeln(childGTIN);

            if(childGTIN != "" && sParentGTIN != "")
            {
                var tradeItemLink = getTradeItemLink(sParentGTIN,
childGTIN);
                if(tradeItemLink != null)
                {
                    tradeItemLink.setLinkAttribute("Quantity", "0");

                    if(status != "")
                    {
                        tradeItemLink.setLinkAttribute("Status", status);
                    }
                    else
                    {
                        var currentStatus =
tradeItemLink.getLinkAttributeValue("Status");
                        var toStatus =
checkString(getToStatus(currentStatus, false), "");
                        if(toStatus != "")
                        {
                            tradeItemLink.setLinkAttribute("Status", toStatus);
                        }
                    }

                    var linkSaved =
tradeItemLink.saveTradeItemLink();

                    logFile.writeln("=====");
                    logFile.writeln("TradeItem Link
"+sParentGTIN+" <==>"+childGTIN+" Saved");
                    logFile.writeln("====="+linkSaved+"=====
=====");

                    if(txn != "" && txn != null)
                    {
                        logFile.writeln("=====");
                        logFile.writeln("Creating
Transaction");
                        var bCreateTxn =
createTransaction(tradeItemLink, txn, "master");

```

```

//
    logFile.writeln((bCreateTxn == true) ? "Done":"Failure");

    logFile.writeln("====="+bCreateTxn+"=====
=====");
    }

    if(notf != "" && notf != null)
    {

        logFile.writeln("=====");
        logFile.writeln("Creating
Notification");
        var bCreateNotf =
createNotification(tradeItemLink,notf,"master");
        //
        logFile.writeln((bCreateNotf == true) ? "Done":"Failure");

        logFile.writeln("====="+bCreateNotf+"=====
=====");
    }
    }
}

}

}

logFile.save("GDS Sample Import - Update TradeItem Links");

```

### Auto-update the Status of Trade Item Links after Import

Trade Item Links can be updated with the auto status functionality. It means that if a trade item link is being updated and the status is not set in the xml file, the script can correctly set the status of the trade item link based on its existing status in the system. The following table shows the rules that govern auto status of trade item links. In the sample script shown above, a lookup table with the name TradingItemLink\_AutoStatus\_Lookup is referenced which is the same as shown.

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
Synchronized	Modified Locally	MODIFIED_ITEM_LINK_APPROVED	MODIFIED_ITEM_LINK_APPROVED
Modified Locally	Modified Locally		

### Import Trading Partners

```

logFile = createOtherOut("GDS Sample TradingPartners Import - New
TradingPartners");

var sGLN = "";
var sTPName = "";
var sCountry = "";
var sGLNID = "";
var status = "";
var tpSaved = false;

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample
TradingPartners Import - New TradingPartners/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);

forEachXMLNode("TradingPartner/")
{
    sGLN = checkString(parseXMLNode("Attribs/GlobalLocationNumber"), "");
    sTPName = checkString(parseXMLNode("Attribs/TradingPartnerName"), "");
    sCountry = checkString(parseXMLNode("Attribs/Country"), "");
    sGLNID = checkString(parseXMLNode("Attribs/GLNIdentifier"), "");
    status = checkString(parseXMLNode("Attribs/Status"), "");

    if(sGLN != "" && sTPName != "" && sCountry != "" && sGLNID != "" &&
status != "")
    {
        var tradingPartner = getTradingPartner(sGLN);

        if(tradingPartner == null)
        {
            tradingPartner = new
TradingPartner(sGLN,sTPName,sCountry,sGLNID);
            if(status != null && status != "")
            {
                tradingPartner.setPartnerAttribute("GLN_Spec/Status",status);
            }

            tpSaved = tradingPartner.saveTradingPartner();

            logFile.writeln("=====");
            logFile.writeln("Trading Partner Saved "+sGLNs);

            logFile.writeln("====="+tpSaved+"=====
=====");

            logFile.writeln("=====");
            logFile.writeln("Creating Transaction");
            var bCreateTxn =
createTransaction(tradingPartner,"NEW_TRADING_PARTNER_APPROVED","master");
            // logFile.writeln((bCreateTxn == true) ?
"Done":"Failure");

            logFile.writeln("====="+bCreateTxn+"=====
=====");

            logFile.writeln("=====");

```



```

        logFile.writeln("Creating Notification");
        var bCreateNotf =
createNotification(tradingPartner,"NEW TRADING PARTNER APPROVED","master");
        //          logFile.writeln((bCreateNotf == true) ?
"Done":"Failure");

        logFile.writeln("====="+bCreateNotf+"=====
=====");

        }
        else
        {
            logFile.writeln("Trading Partner "+sGLN+" already
exists");
        }
    }
    else
    {
        logFile.writeln("Some Mandatory Attribute is missing");
    }
}
logFile.save("GDS Sample TradingPartners Import - New TradingPartners");

```

## Sample XML

The Sample XML file for TradingPartners is given below

```

<TrigoXML>
<Header>
<Transaction_Id></Transaction_Id>
<File_Id>01</File_Id>
<Record_Id>01</Record_Id>
<Creation_Date>08092004</Creation_Date>
<Creation_Time>114441</Creation_Time>
</Header>
<TradingPartner>
<Attribs>
<GlobalLocationNumber>4444444444772</GlobalLocationNumber>
<TradingPartnerName>test</TradingPartnerName>
<GLNIdentifier>Retailer</GLNIdentifier>
<Country>Egypt</Country>
<Status>Draft</Status>
</Attribs>
</TradingPartner>
<TradingPartner>
<Attribs>
<GlobalLocationNumber>4444576777779</GlobalLocationNumber>
<TradingPartnerName>test ff</TradingPartnerName>
<GLNIdentifier>Manufacturer</GLNIdentifier>
<Country>Egypt</Country>
<Status>Draft</Status>
</Attribs>
</TradingPartner>
</TrigoXML>

```

## Import Trading Partners in a Particular Status

```

function getToStatus(strStatus,checkCompliance)
{
    var setStatus = "";
    var tradingPartnerStatusSet =
tradingPartnerStatusLkp.getCtgItemByAttributeValue("TradingPartner_AutoStatus
_Lookup_Spec/From_Status",strStatus);
    if(tradingPartnerStatusSet != null)
    {
        forEachItemSetElement(tradingPartnerStatusSet, statusItem)
        {
            var toStatus =
checkString(statusItem.getCtgItemAttrib("TradingPartner_AutoStatus_Lookup_Spe
c/To_Status"), "");
            if(toStatus != "")
            {
                setStatus = toStatus;
                notf =
checkString(statusItem.getCtgItemAttrib("TradingPartner_AutoStatus_Lookup_Spe
c/Notification"), "");
                txn =
checkString(statusItem.getCtgItemAttrib("TradingPartner_AutoStatus_Lookup_Spe
c/Transaction"), "");
                break;
            }
        }
    }
    logFile.writeln(setStatus);
    return setStatus;
}

logFile = createOtherOut("GDS Sample Import - Update Trading Partner.txt");
var tradingPartnerStatusLkp =
getCtgByName("TradingPartner_AutoStatus_Lookup");

var sGLN = "";
var notf = "";
var txn = "";

var docFiles = getDocStoreFilesInDirectory("/feed_files/GDS Sample
TradingPartners Import - Update TradingPartners With AutoStatus/");
docFiles = docFiles.sort();

var xmlFilePath = docFiles[docFiles.size() - 1];
var xmlFile = getDocByPath(xmlFilePath);
var xmlDoc = new XmlDocument(xmlFile);
var hmAttributePathsToValues = [];
var tpSaved = false;

specMap = getSpecMapByName("GDS XML GLN Attributes Spec Map");
if(specMap != null)
{
    forEachXMLNode("TradingPartner/")
    {
        var hmDstPathsToSrcPaths = [];
        var hmDstPathsToSrcPaths = specMap.getMappings();

        forEachHmElement(hmDstPathsToSrcPaths, dstPath, srcPath)
        {
            if(srcPath.endsWith("GlobalLocationNumber"))
            {

```

```

        sGLN =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
    }
    else
    {
        var value =
checkString(parseXMLNode(srcPath.getRidOfRootName().getRidOfRootName()), "");
        if(value != "")
        {
            hmAttributePathsToValues[dstPath] = value;
        }
    }
}

if(sGLN != null || sGLN != "")
{
    var tradingPartner = getTradingPartner(sGLN);
    if(tradingPartner != null)
    {
        if(hmAttributePathsToValues != null &&
hmAttributePathsToValues.size() != 0)
        {
            tradingPartner.setPartnerAttributes(hmAttributePathsToValues);
        }

        var xmlTPStatus =
checkString(hmAttributePathsToValues["GLN_Spec/Status"], "");

        if(xmlTPStatus == "")
        {
            var tpStatus =
checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Status"), "");

            if(tpStatus != "")
            {
                var status =
checkString(getToStatus(tpStatus, false), "");
                if(status != "")
                {
                    tradingPartner.setPartnerAttribute("GLN_Spec/Status", status);
                }
            }
        }

        tpSaved = tradingPartner.saveTradingPartner();
    }

    logFile.writeln("=====");
    logFile.writeln("Trading Partner Saved");

    logFile.writeln("====="+tpSaved+"=====
==");

    if(txn != "" && txn != null)
    {

```

```

        logFile.writeln("=====");
        logFile.writeln("Creating Transaction");
        var bCreateTxn =
createTransaction(tradingPartner,txn,"master");
        //          logFile.writeln((bCreateTxn == true)
? "Done":"Failure");

        logFile.writeln("===== "+bCreateTxn+"=====
=====");
    }

    if(notf != "" && notf != null)
    {

        logFile.writeln("=====");
        logFile.writeln("Creating Notification");
        var bCreateNotf =
createNotification(tradingPartner,notf,"master");
        //          logFile.writeln((bCreateNotf == true)
? "Done":"Failure");

        logFile.writeln("===== "+bCreateNotf+"=====
=====");
    }
}

}

}

logFile.save("GDS Sample Import - Update Trading Partner.txt");

```

### Auto-update the Status of the Trading Partner after Import

Trading Partners can be updated with the auto status functionality. It means that if a trading partner is being updated and the status is not set in the xml file, the script can correctly set the status of the trading partner based on its existing status in the system. The following table shows the rules that govern auto status of trading partners. In the sample script shown above a lookup table with the name TradingPartner AutoStatus Lookup is referenced which is the same as shown.

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

### Catalog to Catalog Import

For information about catalog to catalog import, refer the *WebSphere Product Version 5.1.0.3 User Guide*.

## Notifications after Import

Most of the scripts above demonstrate how notifications can be created using createNotification() script operation.

## For Export

The various scenarios for export are:

- [Export Trade Items that have changed after some day or newly synched items](#)
- [Export Trade Items which are in a Particular Status](#)
- [Export Trade Items which belong to a Particular Category](#)
- [Export Few Attributes of a Trade Item](#)
- [Export Based on Certain Conditions](#)
- [Export Trade Items Based on Local Status](#)
- [Export All Trade Item Links](#)
- [Export Trade Item Links Based on Status](#)
- [Export All Trading Partners](#)
- [Export Trading Partners – Search Based](#)

### Export Trade Items that have changed after some day or newly synched items

```
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

var txnType = inputs["Transaction_Type"];
var startDateStr = inputs["StartDate"];
var endDateStr = inputs["EndDate"];

var startDate = new Date("dd-MMM-yyyy HH:mm",startDateStr);
var endDate = new Date("dd-MMM-yyyy HH:mm",endDateStr);

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

libScript.getFunctionByName("generateXMLHeader").invoke(out);

if(tradeItems != null && tradeItems.size() != 0)
{
    for(iVar = 0; iVar < tradeItems.size(); iVar++)
    {
        libScript.getFunctionByName("generateXMLForTradeItem").invoke(tradeItems[iVar],out);
    }
}
libScript.getFunctionByName("generateXMLFooter").invoke(out);
```

## Export Trade Items which are in a Particular Status

```
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");
catchError(oErr)
{
    var hmSearchKeysToValues = [];
    forEachHmElement(inputs, searchKey, searchValue)
    {
        if (checkString(searchValue, "") != "")
        {
            hmSearchKeysToValues[searchKey] = searchValue;
        }
    }

    var tradeItems = [];
    tradeItems = searchTradeItems(hmSearchKeysToValues);

    libScript.getFunctionByName("generateXMLHeader").invoke(out);

    if (tradeItems != null && tradeItems.size() != 0)
    {
        for (iVar = 0; iVar < tradeItems.size(); iVar++)
        {
            libScript.getFunctionByName("generateXMLForTradeItem").invoke(tradeItems[iVar], out);
        }
    }

    libScript.getFunctionByName("generateXMLFooter").invoke(out);
}
```

## Export Trade Items which belong to a Particular Category

Refer the script given for Export Items which are in a Particular Status.

## Export Few Attributes of a Trade Item

Refer the script given for Export Items which are in a Particular Status.

The Exports scripts above which search for trade items based on status or category make use of an input spec whose xml is given below. Using this input spec, the same level of search functionality that is provided in the GUI can be achieved.

```
<?xml version="1.0" encoding="UTF-8"?>
<TrigoXML>
  <Header>
    <Type>Spec</Type>
    <Action>Create</Action>
    <CompanyName>demand-side</CompanyName>
    <SpecName>GDS Sample Export - Search Input Spec</SpecName>
    <Attribs>
      <Attrib>
        <Name>TYPE</Name>
        <Value>SCRIPT_INPUT_SPEC</Value>
      </Attrib>
    </Attribs>
  </Header>
```

```

<Nodes>
  <Node>
    <Name>EAN_UCC_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
      <Attrib>
        <Name>MAXLENGTH</Name>
        <Value><![CDATA[3000]]></Value>
      </Attrib>
      <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
      </Attrib>
      <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
      </Attrib>
      <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
      </Attrib>
    </Attribs>
  </Node>
  <Node>
    <Name>GTIN</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
      <Attrib>
        <Name>MAXLENGTH</Name>
        <Value><![CDATA[3000]]></Value>
      </Attrib>
      <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
      </Attrib>
      <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
      </Attrib>
      <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
      </Attrib>
    </Attribs>
  </Node>
  <Node>
    <Name>GTIN_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
      <Attrib>
        <Name>MAXLENGTH</Name>
        <Value><![CDATA[3000]]></Value>
      </Attrib>
      <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
      </Attrib>
      <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
      </Attrib>
      <Attrib>
        <Name>TYPE</Name>

```

```

        <Value><![CDATA[STRING]]></Value>
      </Attrib>
    </Attribs>
  </Node>
  <Node>
    <Name>GLOBAL_STATUS</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
      <Attrib>
        <Name>MAXLENGTH</Name>
        <Value><![CDATA[3000]]></Value>
      </Attrib>
      <Attrib>
        <Name>LOOKUP_TABLE</Name>
        <Value><![CDATA[Item_Status_LookUp]]></Value>
      </Attrib>
      <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
      </Attrib>
      <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
      </Attrib>
      <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[LOOKUP_TABLE]]></Value>
      </Attrib>
    </Attribs>
  </Node>
  <Node>
    <Name>BRAND_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
      <Attrib>
        <Name>MAXLENGTH</Name>
        <Value><![CDATA[3000]]></Value>
      </Attrib>
      <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
      </Attrib>
      <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
      </Attrib>
      <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
      </Attrib>
    </Attribs>
  </Node>
  <Node>
    <Name>PRODUCT_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
      <Attrib>
        <Name>MAXLENGTH</Name>
        <Value><![CDATA[3000]]></Value>
      </Attrib>
      <Attrib>
        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
      </Attrib>
    </Attribs>
  </Node>

```



```

        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>DESCRIPTION</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>TARGET_MARKET</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>LOOKUP_TABLE</Name>
            <Value><![CDATA[Target_Market_LookUp]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>LANGUAGE_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>

```

```

        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>LOOKUP_TABLE</Name>
            <Value><![CDATA[Language_LookUp]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>IP_GLN</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>IP_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>

```

```

        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>BRAND_OWNER_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>MANUFACTURER_GLN</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>MANUFACTURER_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
    </Attribs>

```

```

        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>PRODUCT_TYPE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>STRING_ENUMERATION_RULE</Name>
            <Value><![CDATA[var hmProdTypes = []];
var ctg = getCtgByName("Product_Type_LookUp");
var ctgSpecName = ctg.getCatalogSpec().getSpecName();
var itemSet = ctg.getItemSetForCatalog();

forEachItemSetElement(itemSet,item)
{
    var prodTypeDesc = checkString(item.getCtgItemAttrib(ctgSpecName+"/Product
Type Description"), "");
    if(prodTypeDesc != "")
    {
        hmProdTypes.add(prodTypeDesc);
    }
}

res = hmProdTypes;]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING_ENUMERATION]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>UDEX_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
    </Attribs>

```

```

        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>UDEX_CATEGORY_DESC</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
</Nodes>
</TrigoXML> <?xml version="1.0" encoding="UTF-8"?>
<TrigoXML>
    <Header>
        <Type>Spec</Type>
        <Action>Create</Action>
        <CompanyName>demand-qa</CompanyName>
        <SpecName>GDS Sample Export - Search Input Spec</SpecName>
        <Attribs>
            <Attrib>
                <Name>TYPE</Name>
                <Value>SCRIPT_INPUT_SPEC</Value>
            </Attrib>
        </Attribs>
    </Header>
    <Nodes>
        <Node>
            <Name>EAN_UCC_CODE</Name>
            <Type>ATTRIBUTE</Type>
            <Attribs>
                <Attrib>
                    <Name>MAXLENGTH</Name>
                    <Value><![CDATA[3000]]></Value>
                </Attrib>
                <Attrib>
                    <Name>MAX_OCCURRENCE</Name>
                    <Value><![CDATA[1]]></Value>
                </Attrib>
                <Attrib>
                    <Name>MIN_OCCURRENCE</Name>
                    <Value><![CDATA[0]]></Value>
                </Attrib>
                <Attrib>
                    <Name>TYPE</Name>
                    <Value><![CDATA[STRING]]></Value>
                </Attrib>
            </Attribs>
        </Node>
    </Nodes>
</TrigoXML>

```

```

        </Attrib>
      </Attribs>
    </Node>
    <Node>
      <Name>GTIN</Name>
      <Type>ATTRIBUTE</Type>
      <Attribs>
        <Attrib>
          <Name>MAXLENGTH</Name>
          <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
          <Name>MAX_OCCURRENCE</Name>
          <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
          <Name>MIN_OCCURRENCE</Name>
          <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
          <Name>TYPE</Name>
          <Value><![CDATA[STRING]]></Value>
        </Attrib>
      </Attribs>
    </Node>
    <Node>
      <Name>GTIN_NAME</Name>
      <Type>ATTRIBUTE</Type>
      <Attribs>
        <Attrib>
          <Name>MAXLENGTH</Name>
          <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
          <Name>MAX_OCCURRENCE</Name>
          <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
          <Name>MIN_OCCURRENCE</Name>
          <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
          <Name>TYPE</Name>
          <Value><![CDATA[STRING]]></Value>
        </Attrib>
      </Attribs>
    </Node>
    <Node>
      <Name>GLOBAL_STATUS</Name>
      <Type>ATTRIBUTE</Type>
      <Attribs>
        <Attrib>
          <Name>MAXLENGTH</Name>
          <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
          <Name>LOOKUP_TABLE</Name>
          <Value><![CDATA[Item_Status_LookUp]]></Value>
        </Attrib>
        <Attrib>
          <Name>MAX_OCCURRENCE</Name>
          <Value><![CDATA[1]]></Value>
        </Attrib>
      </Attribs>
    </Node>
  </Node>

```

```

        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>BRAND_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>PRODUCT_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>DESCRIPTION</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>

```

```

        <Name>MAX_OCCURRENCE</Name>
        <Value><![CDATA[1]]></Value>
    </Attrib>
    <Attrib>
        <Name>MIN_OCCURRENCE</Name>
        <Value><![CDATA[0]]></Value>
    </Attrib>
    <Attrib>
        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
    </Attrib>
</Attribs>
</Node>
<Node>
    <Name>TARGET_MARKET</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>LOOKUP_TABLE</Name>
            <Value><![CDATA[Target_Market_LookUp]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>LANGUAGE_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>LOOKUP_TABLE</Name>
            <Value><![CDATA[Language_LookUp]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[LOOKUP_TABLE]]></Value>
        </Attrib>
    </Attribs>

```



```

        </Attribs>
    </Node>
    <Node>
        <Name>IP_GLN</Name>
        <Type>ATTRIBUTE</Type>
        <Attribs>
            <Attrib>
                <Name>MAXLENGTH</Name>
                <Value><![CDATA[3000]]></Value>
            </Attrib>
            <Attrib>
                <Name>MAX_OCCURRENCE</Name>
                <Value><![CDATA[1]]></Value>
            </Attrib>
            <Attrib>
                <Name>MIN_OCCURRENCE</Name>
                <Value><![CDATA[0]]></Value>
            </Attrib>
            <Attrib>
                <Name>TYPE</Name>
                <Value><![CDATA[STRING]]></Value>
            </Attrib>
        </Attribs>
    </Node>
    <Node>
        <Name>IP_NAME</Name>
        <Type>ATTRIBUTE</Type>
        <Attribs>
            <Attrib>
                <Name>MAXLENGTH</Name>
                <Value><![CDATA[3000]]></Value>
            </Attrib>
            <Attrib>
                <Name>MAX_OCCURRENCE</Name>
                <Value><![CDATA[1]]></Value>
            </Attrib>
            <Attrib>
                <Name>MIN_OCCURRENCE</Name>
                <Value><![CDATA[0]]></Value>
            </Attrib>
            <Attrib>
                <Name>TYPE</Name>
                <Value><![CDATA[STRING]]></Value>
            </Attrib>
        </Attribs>
    </Node>
    <Node>
        <Name>BRAND_OWNER_NAME</Name>
        <Type>ATTRIBUTE</Type>
        <Attribs>
            <Attrib>
                <Name>MAXLENGTH</Name>
                <Value><![CDATA[3000]]></Value>
            </Attrib>
            <Attrib>
                <Name>MAX_OCCURRENCE</Name>
                <Value><![CDATA[1]]></Value>
            </Attrib>
            <Attrib>
                <Name>MIN_OCCURRENCE</Name>
                <Value><![CDATA[0]]></Value>
            </Attrib>
            <Attrib>

```

```

        <Name>TYPE</Name>
        <Value><![CDATA[STRING]]></Value>
    </Attrib>
</Attribs>
</Node>
<Node>
    <Name>MANUFACTURER_GLN</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>MANUFACTURER_NAME</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>PRODUCT_TYPE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>STRING_ENUMERATION_RULE</Name>
            <Value><![CDATA[var hmProdTypes = []];
var ctg = getCtgByName("Product_Type_LookUp");
var ctgSpecName = ctg.getCatalogSpec().getSpecName();
var itemSet = ctg.getItemSetForCatalog();

```

```

forEachItemSetElement(itemSet,item)
{
    var prodTypeDesc = checkString(item.getCtgItemAttrib(ctgSpecName+"/Product
Type Description"), "");
    if(prodTypeDesc != "")
    {
        hmProdTypes.add(prodTypeDesc);
    }
}

res = hmProdTypes;]]></Value>
</Attrib>
<Attrib>
    <Name>MAX_OCCURRENCE</Name>
    <Value><![CDATA[1]]></Value>
</Attrib>
<Attrib>
    <Name>MIN_OCCURRENCE</Name>
    <Value><![CDATA[0]]></Value>
</Attrib>
<Attrib>
    <Name>TYPE</Name>
    <Value><![CDATA[STRING_ENUMERATION]]></Value>
</Attrib>
</Attribs>
</Node>
<Node>
    <Name>UDEX_CODE</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>
            <Value><![CDATA[0]]></Value>
        </Attrib>
        <Attrib>
            <Name>TYPE</Name>
            <Value><![CDATA[STRING]]></Value>
        </Attrib>
    </Attribs>
</Node>
<Node>
    <Name>UDEX_CATEGORY_DESC</Name>
    <Type>ATTRIBUTE</Type>
    <Attribs>
        <Attrib>
            <Name>MAXLENGTH</Name>
            <Value><![CDATA[3000]]></Value>
        </Attrib>
        <Attrib>
            <Name>MAX_OCCURRENCE</Name>
            <Value><![CDATA[1]]></Value>
        </Attrib>
        <Attrib>
            <Name>MIN_OCCURRENCE</Name>

```

```

        <Value><![CDATA[0]]></Value>
      </Attrib>
    <Attrib>
      <Name>TYPE</Name>
      <Value><![CDATA[STRING]]></Value>
    </Attrib>
  </Attribs>
</Node>
</Nodes>
</TrigoXML>

```

## Export Based on Certain Conditions

Use the script given for Export Items which are in a Particular Status.

### Export Trade Items Based on Local Status

```

var status = checkString(inputs["Status"], "");
var globalLocalCtg = getCtgByName("Global_Local_Catalog");
var globalCtg = getCtgByName("Global_Catalog");
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");
libScript.getFunctionByName("generateXMLHeader").invoke(out);

if(status != "")
{
    var selQuery =
    "[Global_Local_Catalog:Global_Local_Attributes_Spec/GDSItemStatus] =
    "+"\""+status+"\"";
    var sel = new AdvancedSelection(globalLocalCtg, "Test
    Selection", selQuery);
    var itemSet = sel.getItemSetForSelection();

    if(itemSet != null && itemSet.getItemSetSize() != 0)
    {
        forEachItemSetElement(itemSet, item)
        {
            var gtin_tm_ip = item.getCtgItemPrimarykey();
            var gtin_tm_ip_tokens = gtin_tm_ip.parseDelim("_");

            var gtin = gtin_tm_ip_tokens[0];
            var tm = gtin_tm_ip_tokens[1];
            var ip = gtin_tm_ip_tokens[2];
            var eanUccCode =
            globalCtg.getCtgItemByPrimarykey(gtin).getCtgItemAttrib("Global_Attributes_Sp
            ec/tradeItemIdentification/alternateTradeItemIdentification/eanuccCode");
            var tradeItem = getTradeItemDS(eanUccCode, tm, ip, gtin);

            libScript.getFunctionByName("generateXMLForTradeItem").invoke(tradeIte
            m, out);
        }
    }
}

libScript.getFunctionByName("generateXMLFooter").invoke(out);

```

### Export All Trade Item Links

```

var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

// Generate the Header
libScript.getFunctionByName("generateXMLHeader").invoke(out);

```

```

var ctg = getCtgByName("TradeCatalog");
var globalCtg = getCtgByName("Global_Catalog");

// This is For now.....a search can be put in place if there are too many
// items in the catalog
// This contains all gtins that are processed. We can use this to ignore
// variants.
var hmGTINs = [];

var itemSet = ctg.getItemSetForCatalog();
out.writeln("<Links>");
forEachItemSetElement(itemSet,item)
{
    var gtin_tm_ip = item.getCtgItemPrimarykey();
    var gtin_tm_ip_tokens = gtin_tm_ip.parseDelim("_");

    var gtin = gtin_tm_ip_tokens[0];
    var tm = gtin_tm_ip_tokens[1];
    var ip = gtin_tm_ip_tokens[2];
    var eanUccCode =
globalCtg.getCtgItemByPrimarykey(gtin).getCtgItemAttrib("Global_Attributes_Sp
ec/tradeItemIdentification/alternateTradeItemIdentification/eanuccCode");
    var tradeItem = getTradeItemDS(eanUccCode,tm,ip,gtin);

    // Get all the child links
    var childLinks = tradeItem.getTradeItemChildLinks();

    if(childLinks != null && childLinks.size() != 0 &&
!hmGTINs.containsKey(gtin))
    {
        libScript.getFunctionByName("generateXMLForTradeItemLinks").invoke(gti
n,childLinks,out);
        hmGTINs[gtin] = "0";
    }
}
out.writeln("</Links>");
// Generate the Footer
libScript.getFunctionByName("generateXMLFooter").invoke(out);

```

## Export All Trading Partners

```

var ctg = getCtgByName("Trading_Partner_Catalog");
var ctgSpecName = ctg.getCatalogSpec().getSpecName();
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");

libScript.getFunctionByName("generateXMLHeader").invoke(out);

forEachCtgItem(item)
{
    var gln = checkString(item.getCtgItemAttrib(ctgSpecName+"/Global Location
Number (GLN)"), "");
    var tradingPartner = getTradingPartner(gln);
    if(tradingPartner != null)
    {
        libScript.getFunctionByName("generateXMLForTradingPartner").invoke(gln
,tradingPartner,out);
    }
}

```

```
libScript.getFunctionByName("generateXMLFooter").invoke(out);
```

## Export Trading Partners – Search Based

```
var libScript = getScriptByPath("/scripts/triggers/GDSLibrary.GenerateXML");
catchError(oErr)
{
    var hmSearchKeysToValues = [];
    forEachHmElement(inputs, searchKey, searchValue)
    {
        if(checkString(searchValue, "") != "")
        {
            hmSearchKeysToValues[searchKey] = searchValue;
        }
    }

    var tradingPartners = [];
    tradingPartners = searchTradingPartners(hmSearchKeysToValues);

    libScript.getFunctionByName("generateXMLHeader").invoke(out);

    if(tradingPartners != null && tradingPartners.size() != 0)
    {
        for(iVar = 0; iVar < tradingPartners.size(); iVar++)
        {
            var tradingPartner = tradingPartners[iVar];

            libScript.getFunctionByName("generateXMLForTradingPartner").invoke(null, tradingPartner, out);
        }
    }

    libScript.getFunctionByName("generateXMLFooter").invoke(out);
}
```

This script has an input spec whose attributes match the keys of the hashmap that needs to be passed as an argument to the searchTradingPartners() function. This way the search functionality that is provided by the GUI can be stimulated.

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.

```
function generateXMLHeader(xmlFile)
{
    xmlFile.writeln("<TrigoXML>");
    xmlFile.writeln("<Header>");
        xmlFile.writeln("<Transaction_Id></Transaction_Id>");
        xmlFile.writeln("<File_Id>01</File_Id>");
        xmlFile.writeln("<Record_Id>01</Record_Id>");

    xmlFile.writeln("<Creation_Date>" + today().formatDate("ddMMyyyy") + "</Creation_Date>");

    xmlFile.writeln("<Creation_Time>" + today().formatDate("HHmmss") + "</Creation_Time>");
    xmlFile.writeln("</Header>");
}
```

```

}

function generateXMLFooter(xmlFile)
{
    xmlFile.writeln("</TrigoXML>");
}

function generateXMLForTradeItem(tradeItem,xmlFile)
{
    xmlFile.writeln("<Item>");
    xmlFile.writeln("<Attribs>");

    xmlFile.writeln("<GlobalAttribs>");

    xmlFile.writeln("<EANUCCCode>" + checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attributes_Spec/tradeItemIdentification/alternateTradeItemIdentification/eanuccCode"), "") + "</EANUCCCode>");

    xmlFile.writeln("<EANUCCCodeType>" + checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attributes_Spec/tradeItemIdentification/alternateTradeItemIdentification/eanuccType"), "") + "</EANUCCCodeType>");

    xmlFile.writeln("<GlobalTradeItemNumber>" + checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attributes_Spec/GlobalTradeItemNumber"), "") + "</GlobalTradeItemNumber>");

    xmlFile.writeln("<GTINName>" + checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attributes_Spec/GTINName/en_US"), "") + "</GTINName>");
    var item = tradeItem.getItems()[0];

    if(item != null)
    {
        var coreExtTypeCat =
item.getCtgItemCategories("Global_Extension_Hierarchy")[0];
        if(coreExtTypeCat != null)
        {

            xmlFile.writeln("<CoreExtensionType>" + coreExtTypeCat.getPrimaryKey() + "</CoreExtensionType>");

        }
        else
        {

            xmlFile.writeln("<CoreExtensionType></CoreExtensionType>");

        }
    }

    // Format the Product Type
    var sProdType =
checkString(tradeItem.getTradeItemAttributeValueDS("Global_Attributes_Spec/ProductType"), "");

    var sProdTypeAbbr =
getCtgByName("Product_Type_LookUp").getCtgItemByPrimaryKey(sProdType).getCtgItemAttrib("Product_Type_Spec/Product Type Description");

    xmlFile.writeln("<ProductType>" + sProdTypeAbbr + "</ProductType>");

    xmlFile.writeln("<ItemStatus>" + checkString(tradeItem.getTradeItemAttributeValueDS("Global_Local_Attributes_Spec/GDSItemStatus"), tradeItem.getTradeI

```

```

temAttributeValueDS("Global_Attributes_Spec/GDSItemStatus"))+"</ItemStatus>")
;

        xmlFile.writeln("<BrandName>" + checkString(tradeItem.getTradeItemAttribute
uteValueDS("Global_Attributes_Spec/tradeItemInformation/tradeItemDescriptionI
nformation/brandName/en_US"), "") + "</BrandName>");

        xmlFile.writeln("<Pack>" + checkString(tradeItem.getTradeItemAttributeVa
lueDS("Global_Attributes_Spec/Pack"), "") + "</Pack>");

        xmlFile.writeln("<Classification>");

        xmlFile.writeln("<GlobalClassificationCode>" + tradeItem.getItems()[0].g
etCtgItemCategories("UDEX_Hierarchy_UCCNet")[0].getPrimaryKey() + "</GlobalClas
sificationCode>");

        xmlFile.writeln("<GlobalClassificationDescription>" + tradeItem.getItems
()[0].getCtgItemCategories("UDEX_Hierarchy_UCCNet")[0].getDisplayValue(getUse
rLocale()) + "</GlobalClassificationDescription>");

        xmlFile.writeln("<InternalClassificationCode>" + tradeItem.getItems()[0]
.getCtgItemCategories("Internal_Hierarchy")[0].getPrimaryKey() + "</InternalCla
ssificationCode>");

        xmlFile.writeln("<InternalClassificationDescription>" + tradeItem.getIte
ms()[0].getCtgItemCategories("Internal_Hierarchy")[0].getDisplayValue(getUser
Locale()) + "</InternalClassificationDescription>");
        xmlFile.writeln("</Classification>");
        xmlFile.writeln("</GlobalAttribs>");

        xmlFile.writeln("<LocalAttribs>");

        xmlFile.writeln("<TargetMarket>" + checkString(tradeItem.getTradeItemAtt
ributeValueDS("Global_Local_Attributes_Spec/TargetMarket"), "") + "</TargetMarke
t>");

        xmlFile.writeln("<InformationProvider>" + checkString(tradeItem.getTrade
ItemAttributeValueDS("Global_Local_Attributes_Spec/InformationProvider"), "") +
"</InformationProvider>");

        xmlFile.writeln("<InformationProviderName>" + checkString(tradeItem.getT
radeItemAttributeValueDS("Global_Local_Attributes_Spec/InformationProviderNam
e"), "") + "</InformationProviderName>");
        xmlFile.writeln("</LocalAttribs>");

        xmlFile.writeln("</Attribs>");
        xmlFile.writeln("</Item>");
}

function generateXMLForTradeItemLinks(gtin, tradeItemChildLinks, logFile)
{
    forEachHmElement(tradeItemChildLinks, dummy, tradeItemLink)
    {
        logFile.writeln("<Link>");
        logFile.writeln("<Parent>");

        logFile.writeln("<ParentGTIN>" + gtin + "</ParentGTIN>");
        logFile.writeln("</Parent>");

        logFile.writeln("<Child>");

```



```

        logFile.writeln("<ChildGTIN>" + checkString(tradeItemLink.getLinkAttributeValue("GlobalTradeItemNumber"), "") + "</ChildGTIN>");

        logFile.writeln("<Quantity>" + checkString(tradeItemLink.getLinkAttributeValue("Quantity"), "") + "</Quantity>");

        logFile.writeln("<Status>" + checkString(tradeItemLink.getLinkAttributeValue("Status"), "") + "</Status>");
        logFile.writeln("</Child>");
        logFile.writeln("</Link>");
    }
}

function generateXMLForTradingPartner(gln, tradingPartner, logFile)
{
    logFile.writeln("<TradingPartner>");
    logFile.writeln("<Attribs>");

    logFile.writeln("<GlobalLocationNumber>" + checkString(gln, tradingPartner.getPartnerAttributeValue("GLN_Spec/Global Location Number (GLN)")) + "</GlobalLocationNumber>");

    logFile.writeln("<TradingPartnerName>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner Name"), "") + "</TradingPartnerName>");

    logFile.writeln("<GLNIdentifier>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/GLN Identifier"), "") + "</GLNIdentifier>");

    logFile.writeln("<Country>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Trading Partner Country"), "") + "</Country>");

    logFile.writeln("<Status>" + checkString(tradingPartner.getPartnerAttributeValue("GLN_Spec/Status"), "") + "</Status>");
    logFile.writeln("</Attribs>");

    logFile.writeln("</TradingPartner>");
}

```

## 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.

Module	Script Operation Name	Arguments	Description
Notifications	createNotification	IBusinessObject iObj, String notifType, String userName	<ul style="list-style-type: none"> <li>Creates a notification object and returns true if the notification was successful.</li> <li>iObj is the Business Object. The business object can be a trade item or trading partner or and item link.</li> <li>notifType is the notification type. This can be retrieved from the</li> </ul>

Module	Script Operation Name	Arguments	Description
			<p>NotificationSecurityMappingTable lookup table.</p> <ul style="list-style-type: none"> <li>• 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.</li> </ul>
Transaction	createTransaction	IBusinessObject iObj, String txnType, String userName	<ul style="list-style-type: none"> <li>• Creates a transaction object and returns true if the operation was successful.</li> <li>• iObj is the Business Object. The business object can be a trade item or trading partner or and item link.</li> <li>• txnType is the notification type. This can be retrieved from the NotificationSecurityMappingTable lookup table.</li> <li>• 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.</li> </ul>
Trade Item	newTradeItemDS	String sEANUCCCode, String sEANUCCCodeType, String sExtensionName, String sProductType,String sInternalCatCode, [String sGTIN],[String sGTINName],[String sDataPoolCatCode] ]])	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
	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.
	TradeItem.setTradeItemAttributeDS	String attributePath,String attributeValue	Sets the attribute value for the attribute path of the trade item.
	TradeItem.setTradeItemAttributesDS	HashMap hmAttributePathsToValues	Sets the trade item with the given hashmap of attribute paths and their corresponding values.

Module	Script Operation Name	Arguments	Description
			<b>Note:</b> Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
	TradeItem.saveTradeItemDS	NONE	Saves the trade item to the database and returns true if the save was successful.
	searchTradeItemsDS	HashMap hmSearchParamsToValues	<p>Searches and retrieves trade Items 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.</p> <ul style="list-style-type: none"> <li>• EAN_UCC_CODE</li> <li>• GTIN</li> <li>• GTIN_NAME</li> <li>• BRAND_NAME</li> <li>• PRODUCT_NAME</li> <li>• DESCRIPTION</li> <li>• TARGET_MARKET</li> <li>• LANGUAGE_CODE</li> <li>• IP_GLN</li> <li>• IP_NAME</li> <li>• BRAND_OWNER_NAME</li> <li>• MANUFACTURER_GLN</li> <li>• MANUFACTURER_NAME</li> <li>• PRODUCT_TYPE</li> <li>• UDEX_CODE</li> <li>• UDEX_CATEGORY_DESC</li> <li>• GLOBAL_ITEM_STATUS</li> <li>• UNIT_INDICATOR_YES(Value should be YES or NO)</li> <li>• UNIT_INDICATOR_NO(Value should be YES or NO)</li> <li>• UNIT_INDICATOR_NONE(Value should be YES or NO)</li> </ul>
	getTradeItemParentLinks	NONE	Searches for parent trade item links for this trade item.
	getTradeItemChildLinks	NONE	Searches for child trade item links for this trade item.
	getModifiedTradeItems	String transactionType,	Searches and retrieves trade

Module	Script Operation Name	Arguments	Description
		Date startDate, Date endDate	items modified between start date and end date for a given transaction type.
	TradeItem.getTradeItemAttributeValueDS	String sAttrPath	Returns the attribute value of the TradeItem.
	TradeItem.getItems	NONE	Gets the Global and the local item for this TradeItem.
	TradeItem.setUDEXCategoryCode	String code	Sets the trade item with the given UDEX category code.
	TradeItem.setInternalCategoryCode	String code	Sets the trade item with the given internal category code.
	TradeItem.deleteTMIPVariant	NONE	Deletes the TMIP variant of this tradeItem.
	TradeItem.deleteGTIN	NONE	Deletes the GTIN of this tradeItem. This can only be used when the tradeItem does not have any variants.
	getAllTradeItemAttributes	NONE	Returns a hashmap of all tradeItem attributes to their values
TradingPartner	new\$TradingPartner	String sGLN	Creates a new Trading Partner with the given GLN.
	getTradingPartner	String sGLN	Returns a TradingPartner with the given GLN, else returns null.
	TradingPartner.setPartnerAttribute	String attributePath,String attributeValue	Sets the attribute value for the attribute path of the Trading Partner.
	TradingPartner.setPartnerAttributes	HashMap hmAttributePathsToValues	Sets the Trading Partner with the given hashmap of attributepaths and their corresponding values. <b>Note:</b> Multi occurrence attribute paths need to be indexed with #0,#1 and so on in their paths.
	TradingPartner.saveTradingPartner	NONE	Saves the trading partner to the database and returns true if the save was successful.
	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.

Module	Script Operation Name	Arguments	Description
			<ul style="list-style-type: none"> <li>• GLN</li> <li>• PARTNER_NAME</li> <li>• STATUS</li> <li>• CONTACT</li> <li>• PARTY_ROLE</li> <li>• CITY_NAME</li> <li>• COUNTRY_CODE</li> <li>• LANGUAGE</li> </ul>
	TradingPartner.getPartnerAttributeValue	String sAttrPath	Returns the attribute value for the given path in the trade item link.
TradeItemLinks	new\$TradeItemLinks	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. <b>Note:</b> 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. <b>Note:</b> 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:getLinkAttributeValue	String sAttrPath	Returns the attribute value for the given path in the Trade Item Link
	getModifiedTradeItemLinks	TBD	TBD
SpecMap	getMappings	NONE	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","76857635353536","
Test","000000001.000000002.000000012");
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/Pack","8");
    out.writeln(tradeItem);
    out.writeln(tradeItem.saveTradeItemDS());

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

    var notfCreated =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_PENDING","master");
    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/tradeItemInformatio
n/tradeItemDescriptionInformation/brandName/en_US","Coca Cola");
```

```

tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/Pack","8");
tradeItem.setTradeItemAttributeDS("FMCG_Global_Attributes_Spec/tradeItemIdentification/TradeItemIdentificationListType","CA");
tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemStatus","Global Enrichment Completed");

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

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

var notfCreated =
createNotification(tradeItem,"ITEM_GLOBAL_ENRICHMENT_COMPLETED","master");
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","93653453453451","Test",
"000000001.000000002.000000012");
if(tradeItem != null)
{
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/tradeItemInformation/tradeItemDescriptionInformation/brandName/en US","Coca Cola");
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/Pack","8");
    tradeItem.setTradeItemAttributeDS("FMCG_Global_Attributes_Spec/tradeItemIdentification/TradeItemIdentificationListType","CA");
    tradeItem.setTradeItemAttributeDS("Global_Attributes_Spec/GDSItemStatus","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/PartnerSpecificAttri
butes/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.getTradeItemAttributeValueDS("Global_Attributes_Spec/GT
INName/en_US"),"");
    var attr1 =
checkString(tradeItem.getTradeItemAttributeValueDS("Global_Local_Attributes_S
pec/PartnerSpecificAttributes/Attr1"),"");
    var ipName =
checkString(tradeItem.getTradeItemAttributeValueDS("Global_Local_Attributes_S
pec/InformationProviderName"),"");

    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", "master");
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("4564566666677","CHILD","EACH","10000056","000000001.000000002.0
00000012");

var itemSaved = tradeItem.saveTradeItemDS();
out.writeln("TradeItem [4564566666677] Saved ? : "+itemSaved);

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

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

var tradeItemLink = new TradeItemLink("4564566666660","4564566666677",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_PENDING","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 = getTradeItemDS("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 = 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);
```



---

## Appendix A – Preceded Roles, Access Control Groups, and Selections – supply side

This appendix provides information about the various Roles, Access Control Groups (ACGs), and Selections that are preceded in the solution. It also provides the list of accessible features and operations that can be performed by the respective roles.

### **Roles**

For every IBM Global Data Synchronization for WebSphere Product Center user a set of Roles can be assigned. As an example, consider the company XYZ. This company has four main departments such as Lighting, Consumer Electronics, Software, and Components. Each of these departments has a Global Brand Manager. They are:

- Lighting Global Brand Manager
- Consumer Electronics Global Brand Manager
- Software Global Brand Manager
- Components Global Brand Manager

Note: The naming convention is [UDEX CAT SPECIFIC] Global Brand Manager. That is, Lighting is the UDEX specific category.

The Global Brand Managers are responsible for creating a GTIN. Each of these Global Brand Managers has product executives. Once the Global Brand Manager creates the GTIN, a notification is sent to the product executives to enrich the global attributes. The respective product executives are:

- Lighting Global Product Executive
- Consumer Electronics Global Product Executive
- Software Global Product Executive
- Components Global Product Executive

Note: The Global Brand Manager can also enrich some or all the global attributes.

When all the product executives have completed enriching the global attributes, only then is the global enrichment process complete. A notification that the global enrichment for the trade item is complete is sent to the Global Brand Manager.

Now, if company XYZ has business relations with US, Holland, and Singapore, there will be target market specific Brand Managers for each category. They are:

- Lighting US Brand Manager
- Lighting Holland Brand Manager
- Lighting Singapore Brand Manager
- Consumer Electronics US Brand Manager

- Consumer Electronics Holland Brand Manager
- Consumer Electronics Singapore Brand Manager

Similarly there will be target market specific Brand Managers for each category.

The Global Brand Managers will send the notification to these target market specific Brand Managers to enrich the variants. Each of the target market specific Brand Managers has Product Executives. The Product Executives enrich the variant attributes. Once the variant attributes have been enriched, a notification for variant enrichment completion is sent to the brand managers.

The various roles that are preceded in the IBM Global Data Synchronization for WebSphere Product Center Supply solution are:

- WebSphere Product Centre Administrator
- Global Data Synchronization Administrator
- [UDEX CAT SPECIFIC] Global Brand Manager
- [UDEX CAT SPECIFIC] Global Product Executive
- [UDEX CAT - TM SPECIFIC] Brand Manager
- [[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
- [[UDEX CAT - TM - ATTR GROUP SPECIFIC] User
- Links Publisher
- [UDEX CAT - TM SPECIFIC] Variant Publisher
- [TM Specific] Marketing Manager

Note: These roles can be changed based on your requirements.

### WebSphere Product Centre Administrator

The WebSphere Product Centre Administrator has access to the data model. Typically, the administrator is the PSS user.

The operations that can be setup for IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"> <li>• List</li> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Create Variant</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Approve/Reject</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Approve/Reject</li> </ul>	<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> <li>• Item Transactions</li> <li>• Hierarchy Transactions</li> </ul>	<ul style="list-style-type: none"> <li>• Publications</li> <li>• Initial Load</li> <li>• Item Add</li> <li>• Item Link Add</li> </ul>	<ul style="list-style-type: none"> <li>• Run Publication Reports</li> <li>• View Publication Reports</li> <li>• Run Item Reports</li> <li>• View Item</li> </ul>



IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
Variant • Delete Variant • Enrich Global • Enrich Variant • Approve /Reject • Publish New Item • Send Initial Load • Add Item • Synchronize Changes	/Reject	• Add Item Link	• Trading Partner Transactions	• New Item Request • Item Change Request • New Item Hierarchy Request • Item Hierarchy Change Request • New Trading Partner Request • Trading Partner Change Request	Reports • Run Trading Partner Reports • View Trading Partner Reports

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>Catalog Views</li> <li>• View Items</li> <li>• Add Items</li> <li>• Modify Items</li> <li>• Delete Items</li> <li>• Recategorize Items</li> <li>• Summary Items</li> <li>• Export Attributes</li> <li>• Differences</li> <li>• Rollback</li> <li>• Search</li> <li>• Delete</li> <li>• Run Preview Script</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>Hierarchy Views</li> <li>• View Hierarchy Nodes</li> <li>• Add</li> <li>Hierarchy Nodes</li> <li>• Modify Hierarchy Node Attributes</li> <li>• Delete Hierarchy Nodes</li> <li>• Recategorize Hierarchy Nodes</li> <li>• Summary Hierarchy Nodes</li> <li>• Specmap Hierarchy Nodes</li> <li>• Attributes</li> <li>• Rollback</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Edit Rule</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Perform Import</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• View Items</li> <li>• Add Items</li> <li>• Modify Items</li> <li>• Delete Items</li> <li>• Recategorize Items</li> <li>• View Hierarchy Nodes</li> <li>• Add Hierarchy Nodes</li> <li>• Modify Hierarchy Node Attributes</li> <li>• Delete Hierarchy Nodes</li> <li>• Recategorize Hierarchy Nodes</li> <li>• Specmap Hierarchy Nodes</li> </ul>	<ul style="list-style-type: none"> <li>• View Files</li> <li>• Delete Files</li> </ul>

WebSphere Product Center					
Workflow	Collaboration Area	Spec	Screen	Script	Scheduler
<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Checkout Entries</li> </ul>	<ul style="list-style-type: none"> <li>• Modify Spec</li> <li>• Modify</li> </ul>	<ul style="list-style-type: none"> <li>• View</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Modify</li> <li>• Scripts</li> </ul>	<ul style="list-style-type: none"> <li>• View Company Jobs</li> </ul>

		Spec Map			
--	--	----------	--	--	--

### Global Data Synchronization Administrator

The IBM Global Data Synchronization for WebSphere Product Center administrator has access to all the functionalities in IBM Global Data Synchronization for WebSphere Product Center and some of the functionalities in WebSphere Product Centre such as import/export & user management. The IBM Global Data Synchronization for WebSphere Product Center administrator does not have access to the data model.

The IBM Global Data Synchronization for WebSphere Product Center Administrator has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Publish New Item, Send Initial Load, Add Item, Add Link, and Synchronize Changes
- Explore Transactions
- Manage Items (Create GTIN, Create GTIN Variant (that is, target market and Information provider), View/Edit GTIN Global Attributes, View/Edit Variant Attributes, Enrich GTIN Global Attributes, Enrich Variant Attributes, Delete GTIN, and Delete GTIN Variant)
- Approve/Reject Items
- Create/View/Edit/Delete/Approve Trading Partner
- Import/Export Items, Item Links
- User Management & Security

The operations that can be setup for IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Supply Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"> <li>• List</li> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Create</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Approve</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Approve/Reject</li> </ul>	<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> <li>• Item Transactions</li> <li>• Hierarchy Transactions</li> </ul>	<ul style="list-style-type: none"> <li>• Publications</li> <li>• Initial Load</li> <li>• Item Add</li> <li>• Item Link Add</li> </ul>	<ul style="list-style-type: none"> <li>• Run Publication Reports</li> <li>• View Publication Reports</li> <li>• Run Item Reports</li> </ul>

IBM Global Data Synchronization for WebSphere Product Center Supply Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
Variant • Delete Variant • Enrich Global • Enrich Variant • Approve /Reject • Publish New Item • Send Initial Load • Add Item • Synchronize Changes	/Reject	• Add Item Link	Transactions • Trading Partner Transactions	Add • New Item Request • Item Change Request • New Item Hierarchy Request • Item Hierarchy Change Request • New Trading Partner Request • Trading Partner Change Request	• View Item Reports • Run Trading Partner Reports • View Trading Partner Reports

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
• List	• List	• List	• List • Perform Import • Delete	• View Items	

WebSphere Product Center					
Workflow	Collaboration Area	Spec	Screen	Script	Scheduler
			• View	• Create Modify	• View Company

				Scripts	Jobs
--	--	--	--	---------	------

**[UDEX CAT SPECIFIC] Global Brand Manager**

The Global Brand Manager is responsible for creating new trade items (global variant). The Global Brand Manager can also edit local variant, delete an item and its local variant, and has the privilege of approving/disapproving an item. There will be one Global Brand Manager for each category.

The Global Brand Manager has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Explore Transactions
- Manage Items (Create GTIN, View/Edit GTIN Global Attributes, View/Edit Variant Attributes, Delete GTIN, and Delete GTIN Variant)
- Approve/Reject Items
- View Trading Partner
- Reports

The operations that the Global Brand Manager can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports

- List
- Create
- Edit
- View
- Modify

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports

y

- Delete
- Delete
- Variant
- Explore
- Approve / Reject

- View
- List
- Modify

- A • C
- p re
- pr at
- o e
- v
- e/
- R
- ej
- ec
- t

- Delete
- Explore

- Edit
- View
- List

<ul style="list-style-type: none"><li>• Modify</li><li>• Delete</li><li>• Approve /Reject</li></ul>	<ul style="list-style-type: none"><li>• Item Transactions</li><li>• Hierarchy Transactions</li></ul>	<ul style="list-style-type: none"><li>• New Item Request<ul style="list-style-type: none"><li>• Item Change Request</li></ul></li><li>• New Item Hierarchy Request<ul style="list-style-type: none"><li>• Item Hierarchy Change Request</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Run Item Reports</li><li>• View Item Reports</li></ul>		
WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	

### [UDEX CAT Specific] Global Product Executive

The Global Product Executive can enrich GTIN global attributes and also view or edit the variant attributes.

The Global Product Executive has access to the following:

- Notifications



- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Manage Items (View/Edit GTIN Global Attributes, View/Edit Variant Attributes, and Enrich GTIN Global Attributes)
- View Trading Partner

The operations that the Global Product Executive can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Explore</li> <li>• Enrich Global</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Delete</li> </ul>		<ul style="list-style-type: none"> <li>• New Item Request</li> </ul>	

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>		<ul style="list-style-type: none"> <li>• View Items</li> </ul>	

### [UDEX CAT – TM Specific] Brand Manager

The [UDEX CAT – TM Specific] Brand Manager can create, enrich, delete a variant of an item., approve, or disapprove the variant..

The Brand Manager has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Explore Transactions
- Manage Items (Create GTIN Variant, View GTIN Global Attributes, View/Edit Variant Attributes, Enrich Variant Attributes, and Delete GTIN Variant)
- Approve/Disapprove GTIN Variant

The operations that the Brand Manager can perform in IBM Global Data Synchronization for WebSphere Product Center and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"><li>• List</li><li>• Edit</li><li>• View</li><li>• Modify</li><li>• Explore</li><li>• Create Variant</li><li>• Enrich Variant</li><li>• Delete Variant</li><li>• Approve/Reject</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• View</li></ul>	<ul style="list-style-type: none"><li>• Item Transactions</li></ul>	<ul style="list-style-type: none"><li>• New Item Request</li><li>• Item Change Request</li></ul>	

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	

### [UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive

The [UDEX CAT – TM – ATTR GROUP SPECIFIC] Product Executive can enrich variant attributes.

The Product Executive has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Manage Items (View GTIN Global Attributes, View/Edit Variant Attributes, and Enrich Variant Attributes)
- View Trading Partner

The operations that the Product Executive can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"><li>• List</li><li>• Edit</li><li>• View</li><li>• Modify</li><li>• Explore</li><li>• Enrich Variant</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li></ul>		<ul style="list-style-type: none"><li>• New Item Request</li></ul>	

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	

### [UDEX CAT - TM - ATTR GROUP SPECIFIC] User

The [UDEX CAT – TM – ATTR GROUP SPECIFIC] User can only view GTIN global attributes, view variant attributes and trading partner information.

The User has access to the following:

- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Manage Items (View GTIN Global Attributes, View Variant Attributes)
- View Trading Partner

The operations that the User can perform in IBM Global Data Synchronization for WebSphere Product Center and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li></ul>			

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	

### Links Publisher

The Links Publisher can publish the links to the data pool and has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Add Link
- Explore Transactions
- Manage Items (View GTIN Global Attributes, View Variant Attributes)
- View Trading Partner

The operations that the Links Publisher can perform in IBM Global Data Synchronization for WebSphere Product Center and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Add Item Link</li> </ul>	<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> </ul>	<ul style="list-style-type: none"> <li>• Publications</li> <li>• Item Link Add</li> </ul>	

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>		<ul style="list-style-type: none"> <li>• View Items</li> </ul>	

#### [UDEX CAT - TM SPECIFIC] Variant Publisher

The [UDEX CAT - TM SPECIFIC] Variant Publisher can publish the variants to the data pool and has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Publish New Item, Send Initial Load, Add Item, Synchronize Changes
- Explore Transactions
- Manage Items (View GTIN Global Attributes, View Variant Attributes)
- View Trading Partner

The operations that the Variant Publisher can perform in IBM Global Data Synchronization for WebSphere Product Center and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> <li>• Publish New Item</li> <li>• Send Initial Load</li> <li>• Add Item</li> <li>• Synchronize Changes</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> </ul>	<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> </ul>	<ul style="list-style-type: none"> <li>• Publications</li> <li>• Initial Load</li> <li>• Item Add</li> </ul>	

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>		<ul style="list-style-type: none"> <li>• View Items</li> </ul>	

### [TM Specific] Marketing Manager

The [TM Specific] Marketing Manager can create, view, edit, delete, approve or reject a Trading Partner for a specific target market.

The Marketing Manager has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Explore Transactions
- Manage Items (View GTIN Global Attributes, View Variant Attributes)
- Create/View/Edit/Delete/Approve Trading Partner

The operations that the Marketing Manager can perform in IBM Global Data Synchronization for WebSphere Product Center and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution					
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications	Reports
<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Approve /Reject</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> </ul>	<ul style="list-style-type: none"> <li>• Trading Partner Transactions</li> </ul>	<ul style="list-style-type: none"> <li>• New Trading Partner Request</li> <li>• Trading Partner Change Request</li> </ul>	<ul style="list-style-type: none"> <li>• Run Trading Partner Reports</li> <li>• View Trading Partner Reports</li> </ul>

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>		<ul style="list-style-type: none"> <li>• View Items</li> </ul>	

## Workflow Roles

The following workflow roles are preceded in the solution. The user needs to have these roles to do actions related to workflows. When creating new users to do some action(s) in a specified workflow, then that user should be assigned with a specific role(s).

- GDS\_Approve\_Item\_WF\_Role  
This role is required to approve an item.
- GDS\_Approve\_Modified\_TP\_WF\_Role  
This role is required to approve a modified Trading Partner.
- GDS\_Approve\_New\_TP\_WF\_Role  
This role is required to approve a new Trading Partner.
- GDS\_Approve\_TP\_Deletion\_WF\_Role  
This role is required to approve a deleted Trading Partner.
- GDS\_Enrich\_Global\_Attributes\_WF\_Role  
This role is required to enrich the global attributes.

- GDS\_Enrich\_Local\_Attributes\_WF\_Role  
This is required to enrich the local attributes.
- GDS\_Modify\_Global\_And\_Check\_Compliance\_WF\_Role  
This is required to change global attributes of a registered item and perform the compliance check.
- GDS\_Approve\_Modify\_Global\_Item\_WF\_Role  
This role is required to approve changes in global attributes of a registered item.
- GDS\_Modify\_Local\_And\_Check\_Compliance\_WF\_Role  
This role is required to change local attributes of a registered item and perform the compliance check.
- GDS\_Approve\_Modify\_Local\_Item\_WF\_Role  
This role is required to approve changes in local attributes of a registered item.
- GDS\_Approve\_Modify\_Link\_WF\_Role  
This role is required to approve links. This may be a new link approval or modify link approval or deletion link approval.

## Access Control Groups

Examples of UDEX Category and Target Market Specific ACGs are provided in the following table.

ACG	Roles	
ALCOHOL BASED MIXERS ACG	ALCOHOL BASED MIXERS Global Brand Manager	[UDEX CAT SPECIFIC] Global Brand Manager
	ALCOHOL BASED MIXERS Global Product Executive	[UDEX CAT SPECIFIC] Global Product Executive
	ALCOHOL BASED MIXERS - US Brand Manager	[UDEX CAT - TM SPECIFIC] Brand Manager
	ALCOHOL BASED MIXERS UK Brand Manager	
	ALCOHOL BASED MIXERS - US - Logistics Product Executive	[[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
	ALCOHOL BASED MIXERS - US - Measures Product Executive	
	ALCOHOL BASED MIXERS - US - Nutrition Product Executive	
	ALCOHOL BASED MIXERS - UK - Logistics Product Executive	
	ALCOHOL BASED MIXERS - UK - Measures Product Executive	
	ALCOHOL BASED MIXERS - UK - Nutrition Product Executive	



ACG	Roles	
US ACG	ALCOHOL BASED MIXERS - US Brand Manager	[UDEX CAT - TM SPECIFIC] Brand Manager
	ALCOHOL BREWING_MAKING KITS US Brand Manager	
	ALCOHOL BASED MIXERS - US - Logistics Product Executive	[[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
	ALCOHOL BASED MIXERS - US - Measures Product Executive	
	ALCOHOL BASED MIXERS - US - Nutrition Product Executive	
	ALCOHOL BREWING_MAKING KITS - US - Logistics Product Executive	
	ALCOHOL BREWING_MAKING KITS - US - Measures Product Executive	
	ALCOHOL BASED MIXERS - US - Logistics User	[[UDEX CAT - TM - ATTR GROUP SPECIFIC] User
	ALCOHOL BASED MIXERS - US - Measures User	
	ALCOHOL BASED MIXERS - US - Nutrition User	
	ALCOHOL BREWING_MAKING KITS - US - Logistics User	
	ALCOHOL BASED MIXERS - US Variant Publisher	[UDEX CAT - TM SPECIFIC] Variant Publisher
	ALCOHOL BREWING_MAKING KITS US Variant Publisher	

## Selections

The following Selections have been pre-seeded into the model.

Selection	ACG	Catalog	Hierarchy	Selected Categories
UNET_ALCOHOL BASED MIXES/MIXERS	Alcohol ACG	Globa_Catalog	UDEX_Hierarchy_UCCNet	<ul style="list-style-type: none"> <li>• ALCOHOL BASED MIXES/MIXERS</li> <li>• ALCOHOL BASED COCKTAILS/MIXES</li> <li>• ALCOHOL BASED MIXERS</li> </ul>
UNET_ANIMAL/PET ACCESSORIES OTHER	Animal Pet ACG	Global_Catalog	UDEX_Hierarchy_UCCNet	<ul style="list-style-type: none"> <li>• ANIMAL/PET ACCESSORIES OTHER</li> <li>• ANIMAL/PET ACCESSORIES OTHER</li> <li>• ANIMAL/PET ACCESSORIES OTHER VARIETY PACKS/SETS</li> </ul>
UNET_CEREAL GRAINS WITH ADDITIONS - DRY (SHELF STABLE)	Cereal Grains ACG	Global_Catalog	UDEX_Hierarchy_UCCNet	<ul style="list-style-type: none"> <li>• CEREAL GRAINS WITH ADDITIONS - DRY (SHELF STABLE)</li> <li>• BULGUR WHEAT WITH ADDITIONS - DRY (SHELF STABLE)</li> <li>• CEREAL GRAINS OTHER WITH ADDITIONS - DRY (SHELF STABLE)</li> </ul>
TM_Africa	Africa ACG	Global_Local_Catalog	Target_Market_Hierarchy	<ul style="list-style-type: none"> <li>• EG</li> <li>• SA</li> </ul>
TM_NorthAmerica	North America ACG	Global_Local_Catalog	Target_Market_Hierarchy	<ul style="list-style-type: none"> <li>• US</li> <li>• CA</li> </ul>

---

## Appendix B – Pre-seeded Roles, Access Control Groups, and Selections –demand side

This appendix provides information about the various Roles, Access Control Groups (ACGs), and Selections that are pre-seeded in the solution. It also provides the list of accessible features and operations that can be performed by the respective roles.

### **Roles**

For every IBM Global Data Synchronization for WebSphere Product Center user a set of Roles can be assigned. As an example, consider the company XYZ. This company has four main departments such as Lighting, Consumer Electronics, Software, and Components. Each of these departments has a Global Brand Manager. They are:

- Lighting Global Brand Manager
- Consumer Electronics Global Brand Manager
- Software Global Brand Manager
- Components Global Brand Manager

Note: The naming convention is [UDEX CAT SPECIFIC] Global Brand Manager. That is, Lighting is the UDEX specific category.

The Global Brand Managers are responsible for creating a GTIN. Each of these Global Brand Managers has product executives. Once the Global Brand Manager creates the GTIN, a notification is sent to the product executives to enrich the global attributes. The respective product executives are:

- Lighting Global Product Executive
- Consumer Electronics Global Product Executive
- Software Global Product Executive
- Components Global Product Executive

Note: The Global Brand Manager can also enrich some or all the global attributes.

When all the product executives have completed enriching the global attributes, only then is the global enrichment process complete. A notification that the global enrichment for the trade item is complete is sent to the Global Brand Manager.

Now, if company XYZ has business relations with US, Holland, and Singapore, there will be target market specific Brand Managers for each category. They are:

- Lighting US Brand Manager
- Lighting Holland Brand Manager
- Lighting Singapore Brand Manager
- Consumer Electronics US Brand Manager

- Consumer Electronics Holland Brand Manager
- Consumer Electronics Singapore Brand Manager

Similarly there will be target market specific Brand Managers for each category.

The Global Brand Managers will send the notification to these target market specific Brand Managers to enrich the variants. Each of the target market specific Brand Managers has Product Executives. The Product Executives enrich the variant attributes. Once the variant attributes have been enriched, a notification for variant enrichment completion is sent to the brand managers.

The various roles that are pre-seeded in the IBM Global Data Synchronization for WebSphere Product Center solution are:

- WebSphere Product Centre Administrator
- Global Data Synchronization Administrator
- [UDEX CAT SPECIFIC] Global Brand Manager
- [UDEX CAT SPECIFIC] Global Product Executive
- Target Market Brand Manager
- Target Market Product Executive
- Marketing Manager
- Marketing Executive
- Data Sync Manager
- Data Sync Executive

Note: These roles can be changed based on your requirements.

### WebSphere Product Centre Administrator

The WebSphere Product Centre Administrator has access to the data model. Typically, the administrator is the PSS user.

The operations that can be setup for IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"> <li>• List</li> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Create Variant</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Approve/Reject</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Approve/Reject</li> </ul>	<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> <li>• Item Transactions</li> <li>• Hierarchy Transactions</li> </ul>	<ul style="list-style-type: none"> <li>• Publications</li> <li>• Initial Load</li> <li>• Item Add</li> <li>• Item Link Add</li> <li>• New Item Request</li> <li>• Item Change Request</li> <li>• New Item Hierarchy Request</li> </ul>

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
Variant • Delete Variant • Enrich Global • Enrich Variant • Approve /Reject • Publish New Item • Send Initial Load • Add Item • Synchronize Changes	/Reject	• Add Item Link	• Trading Partner Transactions	• Item Hierarchy Change Request • New Trading Partner Request Trading Partner Change Request

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>Catalog Views</li> <li>• View Items</li> <li>• Add Items</li> <li>• Modify Items</li> <li>• Delete Items</li> <li>• Recategorize Items</li> <li>• Summary Items</li> <li>• Export Attributes</li> <li>• Differences</li> <li>• Rollback</li> <li>• Search</li> <li>• Delete</li> <li>• Run Preview Script</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>Hierarchy Views</li> <li>• View Hierarchy Nodes</li> <li>• Add</li> <li>Hierarchy Nodes</li> <li>• Modify Hierarchy Node Attributes</li> <li>• Delete Hierarchy Nodes</li> <li>• Recategorize Hierarchy Nodes</li> <li>• Summary Hierarchy Nodes</li> <li>• Specmap Hierarchy Nodes</li> <li>• Attributes</li> <li>• Rollback</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Edit Rule</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Perform Import</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• View Items</li> <li>• Add Items</li> <li>• Modify Items</li> <li>• Delete Items</li> <li>• Recategorize Items</li> <li>• View Hierarchy Nodes</li> <li>• Add Hierarchy Nodes</li> <li>• Modify Hierarchy Node Attributes</li> <li>• Delete Hierarchy Nodes</li> <li>• Recategorize Hierarchy Nodes</li> <li>• Specmap Hierarchy Nodes</li> </ul>	<ul style="list-style-type: none"> <li>• View Files</li> <li>• Delete Files</li> </ul>

WebSphere Product Center					
Workflow	Collaboration Area	Spec	Screen	Script	Scheduler
<ul style="list-style-type: none"> <li>• List</li> <li>• Edit</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Checkout Entries</li> </ul>	<ul style="list-style-type: none"> <li>• Modify Spec</li> <li>• Modify</li> </ul>	<ul style="list-style-type: none"> <li>• View</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Modify</li> <li>• Scripts</li> </ul>	<ul style="list-style-type: none"> <li>• View Company Jobs</li> </ul>

		Spec Map			
--	--	----------	--	--	--

### Global Data Synchronization Administrator

The IBM Global Data Synchronization for WebSphere Product Center administrator has access to all the functionalities in IBM Global Data Synchronization for WebSphere Product Center and some of the functionalities in WebSphere Product Centre such as import/export & user management. The IBM Global Data Synchronization for WebSphere Product Center administrator does not have access to the data model.

The IBM Global Data Synchronization for WebSphere Product Center Administrator has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Publish New Item, Send Initial Load, Add Item, Add Link, and Synchronize Changes
- Explore Transactions
- Manage Items (Create GTIN, Create GTIN Variant (that is, target market and Information provider), View/Edit GTIN Global Attributes, View/Edit Variant Attributes, Enrich GTIN Global Attributes, Enrich Variant Attributes, Delete GTIN, and Delete GTIN Variant)
- Approve/Reject Items
- Create/View/Edit/Delete/Approve Trading Partner
- Import/Export Items, Item Links
- User Management & Security

The operations that can be setup for IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"> <li>• List</li> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Create Variant</li> <li>• Delete Variant</li> <li>• Enrich</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Explore</li> <li>• Approve/Reject</li> </ul>	<ul style="list-style-type: none"> <li>• Create</li> <li>• Edit</li> <li>• View</li> <li>• List</li> <li>• Modify</li> <li>• Delete</li> <li>• Approve/Reject</li> <li>• Add Item Link</li> </ul>	<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> <li>• Item Transactions</li> <li>• Hierarchy Transactions</li> <li>• Trading Partner Transactions</li> </ul>	<ul style="list-style-type: none"> <li>• Publications</li> <li>• Initial Load</li> <li>• Item Add</li> <li>• Item Link Add</li> <li>• New Item Request</li> <li>• Item Change Request</li> <li>• New Item Hierarchy Request</li> <li>• Item Hierarchy Change</li> <li>• New Trading Partner Request</li> <li>• Trading Partner Change</li> </ul>

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
Global <ul style="list-style-type: none"> <li>• Enrich Variant</li> <li>• Approve /Reject</li> <li>• Publish New Item</li> <li>• Send Initial Load</li> <li>• Add Item</li> <li>• Synchronize Changes</li> </ul>				Request

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• Perform Import</li> <li>• Delete</li> </ul>	<ul style="list-style-type: none"> <li>• View Items</li> </ul>	

WebSphere Product Center					
Workflow	Collaboration Area	Spec	Screen	Script	Scheduler
			<ul style="list-style-type: none"> <li>• View</li> </ul>	<ul style="list-style-type: none"> <li>• Create Modify Scripts</li> </ul>	<ul style="list-style-type: none"> <li>• View Company Jobs</li> </ul>



## [UDEX CAT SPECIFIC] Global Brand Manager

The Global Brand Manager is responsible for creating new trade items (global variant). The Global Brand Manager can also edit local variant, delete an item and its local variant. There will be one Global Brand Manager for each category.

The Global Brand Manager has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Explore Transactions
- Manage Items (Create GTIN, View/Edit GTIN Global Attributes, View/Edit Variant Attributes, Delete GTIN, and Delete GTIN Variant)
- View Trading Partner

The operations that the Global Brand Manager can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"><li>• List</li><li>• Create</li><li>• Edit</li><li>• View</li><li>• Modify</li><li>• Delete</li><li>• Delete Variant</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• View</li><li>• List</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• Create</li><li>• Edit</li><li>• View</li><li>• List</li><li>• Modify</li><li>• Delete</li><li>• Approve/Reject</li></ul>	<ul style="list-style-type: none"><li>• Item Transactions</li><li>• Hierarchy Transactions</li><li>• Trading Partner Transactions</li></ul>	<ul style="list-style-type: none"><li>• New Item Request</li><li>• Item Change Request</li><li>• New Item Hierarchy Request</li><li>• Item Hierarchy Change Request</li></ul>

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	<ul style="list-style-type: none"><li>• View Files</li><li>• Delete Files</li></ul>

### [UDEX CAT Specific] Global Product Executive

The Global Product Executive can enrich GTIN global attributes and also view or edit the variant attributes.

The Global Product Executive has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Manage Items (View/Edit GTIN Global Attributes, View/Edit Variant Attributes, and Enrich GTIN Global Attributes)
- View Trading Partner

The operations that the Global Product Executive can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"><li>• List</li><li>• Edit</li><li>• View</li><li>• Modify</li><li>• Explore</li><li>• Enrich Global</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li></ul>	<ul style="list-style-type: none"><li>• Item Transaction</li></ul>	New Item Request

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li><li>•</li></ul>	<ul style="list-style-type: none"><li>• View Files</li><li>• Delete Files</li></ul>

## Target Market Brand Manager

The [UDEX CAT – TM Specific] Brand Manager can create, delete a variant of an item, approve, or disapprove the variant.

The Brand Manager has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, Trade Item Links, and Trading Partners
- Explore Transactions
- Manage Items (Create GTIN Variant, View GTIN Global Attributes, View/Edit Variant Attributes, Enrich Variant Attributes, and Delete GTIN Variant)
- Approve/Disapprove GTIN Variant

The operations that the Target Market Brand Manager can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"><li>• List</li><li>• Edit</li><li>• View</li><li>• Modify</li><li>• Explore</li><li>• Create Variant</li><li>• Delete Variant</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Edit</li><li>• Delete</li><li>• Create</li><li>• Modify</li></ul>	<ul style="list-style-type: none"><li>• Item Transactions</li><li>• Hierarchy Transactions</li></ul>	<ul style="list-style-type: none"><li>• New Item Request</li><li>• Item Change Request</li><li>• New Item Hierarchy Request</li></ul> Item Hierarchy Change Request

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	<ul style="list-style-type: none"><li>• View Files</li><li>• Delete Files</li></ul>

## Target Market Product Executive

The Target Market Product Executive can enrich variant attributes.

The Target Market Product Executive has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, and Trading Partner
- Manage Items (View GTIN Global Attributes, View/Edit Variant Attributes, and Enrich Variant Attributes)
- View Trading Partner

The operations that the Target Market Product Executive can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"><li>• List</li><li>• Edit</li><li>• View</li><li>• Modify</li><li>• Explore</li><li>• Enrich Variant</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>			<ul style="list-style-type: none"><li>• New Item Request</li><li>• Item Change Request</li></ul>

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	View Files Delete Files

## Marketing Manager

The Marketing Manager can view trade Items and trading partner information.

The Marketing Manager has access to the following:

- Notifications
- Quick Search, Basic Search, and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Manage Items (View GTIN Global Attributes, View Variant Attributes)
- View Trading Partner

The operations that the Marketing Manager can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Explore</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li><li>• Edit</li><li>• Delete</li><li>• Create</li><li>• Modify</li><li>• Explore</li><li>• Approve /Reject</li></ul>	<ul style="list-style-type: none"><li>• List</li><li>• View</li></ul>	Trading Partner Transactions	New Trading Partner Request Trading Partner Change Request

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>	<ul style="list-style-type: none"><li>• List</li></ul>		<ul style="list-style-type: none"><li>• View Items</li></ul>	View Files Delete Files

## Marketing Executive

The Marketing Executive can view Trade Items and Trading Partner Information

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items, Trade Item Links, and Trading Partner
- Add Link

- Explore Transactions
- Manage Items (View GTIN Global Attributes, View Variant Attributes)
- View Trading Partner

The operations that the Marketing Executive can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> <li>• Edit</li> <li>• Delete</li> <li>• Create</li> <li>• Modify</li> <li>• Explore</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> <li>• View</li> </ul>	<ul style="list-style-type: none"> <li>• Trading Partner Transactions</li> </ul>	<ul style="list-style-type: none"> <li>New Trading Partner Request</li> <li>Trading Partner Change Request</li> </ul>

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>		<ul style="list-style-type: none"> <li>• View Items</li> </ul>	<ul style="list-style-type: none"> <li>View Files</li> <li>Delete Files</li> </ul>

### Data Sync Manager

The Data Sync Manager can process and approve Subscriptions and has access to the following:

- Notifications
- Quick Search, Basic Search , and Advanced Search for Trade Items
- Publish New Item, Send Initial Load, Add Item, Synchronize Changes, Request for Publication, Process Initial Load and Process Item Change and Item Delist
- Explore Transactions

The operations that the Data Sync Manager can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"> <li>• Request For Publication               <ul style="list-style-type: none"> <li>• Stop Synchronization</li> </ul> </li> <li>• Process Initial Load</li> <li>• Process New Item</li> <li>• Process Item Change</li> <li>• Process Item Delist</li> <li>• Process item withdraw</li> <li>• Submit For Enrichment</li> <li>• Classify Item</li> <li>• Synchronize Internally</li> <li>• Publication Explore</li> <li>• Query Item</li> </ul>			<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> </ul>	<ul style="list-style-type: none"> <li>• New Item Publications</li> <li>• Initial Load Publications</li> <li>• Item Change Publication</li> <li>• Item Delist Publication</li> <li>• Item Withdraw Publication               <ul style="list-style-type: none"> <li>• Subscription Request</li> <li>• Filter Request</li> </ul> </li> </ul>

WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>	<ul style="list-style-type: none"> <li>• List</li> </ul>		<ul style="list-style-type: none"> <li>• View Items</li> </ul>	View Files Delete Files

### Data Synch Executive

The Data Synch Executive can process Subscriptions and Filters.

The Data Synch Executive has access to the following:

- Notifications
- Publish New Item, Send Initial Load, Add Item, Synchronize Changes, Request for Publication, Process Initial and Query Items
- Explore Transactions

The operations that the Marketing Manager can perform in IBM Global Data Synchronization for WebSphere Product Center solution and WebSphere Product Center are provided in the table.

IBM Global Data Synchronization for WebSphere Product Center Solution				
Trade Item	Trading Partner	Trade Item Link	Transactions	Notifications
<ul style="list-style-type: none"> <li>• Request For Publication               <ul style="list-style-type: none"> <li>• Stop Synchronization</li> </ul> </li> <li>• Process Initial Load</li> <li>• Process New Item               <ul style="list-style-type: none"> <li>• Process Item Change</li> </ul> </li> <li>• Process Item Delist               <ul style="list-style-type: none"> <li>• Process item withdraw</li> <li>• Submit For Enrichment</li> </ul> </li> <li>• Classify Item</li> <li>• Synchronize Internally</li> <li>• Publication Explore               <ul style="list-style-type: none"> <li>• Synchronize Changes</li> <li>• Classify Item</li> <li>• Synchronize Internally</li> <li>• Query Item</li> </ul> </li> </ul>			<ul style="list-style-type: none"> <li>• Publish Transactions</li> <li>• View Generated XML</li> </ul>	<ul style="list-style-type: none"> <li>• New Item Publications</li> <li>• Initial Load Publications</li> <li>• Item Change Publication</li> <li>• Item Delist Publication</li> <li>• Item Withdraw Publication               <ul style="list-style-type: none"> <li>• Subscription Request</li> <li>• Filter Request</li> </ul> </li> </ul>



WebSphere Product Center					
Catalog	Hierarchy	Selection	Import	Selection Members	DocStore
• List	• List	• List		• View Items	View Files Delete Files

## Workflow Roles

The following workflow roles are pre-seeded in the solution. The user needs to have these roles to do actions related to workflows. When creating new users to do some action(s) in a specified workflow, then that user should be assigned with a specific role(s).

- Administrators
- GDS\_ALL\_WF\_Role
- GDS\_Approve\_Filter\_Deletion\_WF\_Role  
This role is required to approve filter deletions.
- GDS\_Approve\_Filter\_WF\_Role  
This role is required to approve filter creations.
- GDS\_Approve\_Item\_Deletion\_WF\_Role  
This role is required to approve item deletions.
- GDS\_Approve\_Item\_WF\_Role  
This role is required to approve Item creations.
- GDS\_Approve\_Link\_Deletion\_WF\_Role  
This role is required to approve Item Link deletions.
- GDS\_Approve\_Link\_WF\_Role  
This role is required to approve Link creation.
- GDS\_Approve\_Modified\_Filter\_WF\_Role  
This role is required to approve filter modifications.
- GDS\_Approve\_Modified\_TP\_WF\_Role  
This role is required to approve trading partner modifications.
- GDS\_Approve\_Modify\_Link\_WF\_Role  
This role is required to approve Item Link modifications.
- GDS\_Approve\_Modify\_Trade\_Item\_WF\_Role  
This role is required to approve trade item modifications.

- GDS\_Approve\_New\_TP\_WF\_Role  
This role is required to approve creation of new trading partners.
- GDS\_Approve\_Subscription\_Deletion\_WF\_Role  
This role is required to approve subscription deletion.
- GDS\_Approve\_Subscription\_WF\_Role  
This role is required to approve subscription creation.
- GDS\_Approve\_TP\_Deletion\_WF\_Role  
This role is required to approve trading partner deletion.
- GDS\_Enrich\_Global\_Attributes\_WF\_Role  
This role is required to enrich global attributes for a trade item.
- GDS\_Enrich\_Local\_Attributes\_WF\_Role  
This role is required to enrich Local Attributes for a trade item.
- GDS\_Internal\_Sync\_WF\_Role  
This role is required to perform internal synchronization of a trade item.
- GDS\_Modify\_Filter\_WF\_Role  
This role is required to modify a filter.
- GDS\_Modify\_Trade\_Item\_WF\_Role  
This role is required to modify a trade item.
- GDS\_Process\_Initial\_Load\_Publication\_WF\_Role  
This role is required to process an initial load publication.
- GDS\_Process\_Item\_Change\_Publication\_WF\_Role  
This role is required to process an item change publication.
- GDS\_Process\_Item\_Delist\_Publication\_WF\_Role  
This role is required to process an item delist publication.
- GDS\_Process\_Item\_Withdraw\_Publication\_WF\_Role  
This role is required to process an item withdraw publication.
- GDS\_Process\_New\_Item\_Publication\_WF\_Role  
This role is required to process new item publication.
- GDS\_Set\_Category\_Mapping\_WF\_Role  
This role is required to set category mapping
- GDS\_Submit\_For\_Enrichment\_WF\_Role  
This role is required to submit a trade item for enrichment

## Access Control Groups

Examples of UDEX Category and Target Market Specific ACGs are provided in the following table.

ACG	Roles	
ALCOHOL BASED MIXERS ACG	ALCOHOL BASED MIXERS Global Brand Manager	[UDEX CAT SPECIFIC] Global Brand Manager
	ALCOHOL BASED MIXERS Global Product Executive	[UDEX CAT SPECIFIC] Global Product Executive
	ALCOHOL BASED MIXERS - US Brand Manager	[UDEX CAT - TM SPECIFIC] Brand Manager
	ALCOHOL BASED MIXERS UK Brand Manager	
	ALCOHOL BASED MIXERS - US - Logistics Product Executive	[[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
	ALCOHOL BASED MIXERS - US - Measures Product Executive	
	ALCOHOL BASED MIXERS - US - Nutrition Product Executive	
	ALCOHOL BASED MIXERS - UK - Logistics Product Executive	
	ALCOHOL BASED MIXERS - UK - Measures Product Executive	
	ALCOHOL BASED MIXERS - UK - Nutrition Product Executive	
US ACG	ALCOHOL BASED MIXERS - US Brand Manager	[UDEX CAT - TM SPECIFIC] Brand Manager
	ALCOHOL BREWING_MAKING KITS US Brand Manager	
	ALCOHOL BASED MIXERS - US - Logistics Product Executive	[[UDEX CAT - TM - ATTR GROUP SPECIFIC] Product Executive
	ALCOHOL BASED MIXERS - US - Measures Product Executive	
	ALCOHOL BASED MIXERS - US - Nutrition Product Executive	
	ALCOHOL BREWING_MAKING KITS - US - Logistics Product Executive	
	ALCOHOL BREWING_MAKING KITS - US - Measures Product Executive	
	ALCOHOL BASED MIXERS - US - Logistics User	[[UDEX CAT - TM - ATTR GROUP SPECIFIC] User
	ALCOHOL BASED MIXERS - US - Measures User	
	ALCOHOL BASED MIXERS - US - Nutrition User	

ACG	Roles	
	ALCOHOL BREWING_MAKING KITS - US - Logistics User	
	ALCOHOL BASED MIXERS - US Variant Publisher	[UDEX CAT - TM SPECIFIC] Variant Publisher
	ALCOHOL BREWING_MAKING KITS US Variant Publisher	

## Selections

The following Selections have been pre-seeded into the model which means that the following selections will be added to the model, during the installation process of the application.

Selection	ACG	Catalog	Hierarchy	Selected Categories
UNET_ALC OHOL BASED MIXES/MIX ERS	Alcohol ACG	Globa_Catalog	UDEX_Hierarchy_ UCCNet	<ul style="list-style-type: none"> <li>• ALCOHOL BASED MIXES/MIXERS</li> <li>• ALCOHOL BASED COCKTAILS/MIXES</li> <li>• ALCOHOL BASED MIXERS</li> </ul>
UNET_ANI MAL/PET ACCESSORI ES OTHER	Animal Pet ACG	Global_Catalog	UDEX_Hierarchy_ UCCNet	<ul style="list-style-type: none"> <li>• ANIMAL/PET ACCESSORIES OTHER</li> <li>• ANIMAL/PET ACCESSORIES OTHER</li> <li>• ANIMAL/PET ACCESSORIES OTHER VARIETY PACKS/SETS</li> </ul>
UNET_CER EAL GRAINS WITH ADDITIONS - DRY (SHELF STABLE)	Cereal Grains ACG	Global_Catalog	UDEX_Hierarchy_ UCCNet	<ul style="list-style-type: none"> <li>• CEREAL GRAINS WITH ADDITIONS - DRY (SHELF STABLE)</li> <li>• BULGUR WHEAT WITH ADDITIONS - DRY (SHELF STABLE)</li> <li>• CEREAL GRAINS OTHER WITH ADDITIONS - DRY (SHELF STABLE)</li> </ul>
TM_Africa	Africa ACG	Global_Local_Ca talog	Target_Market_Hie rarchy	<ul style="list-style-type: none"> <li>• EG</li> <li>• SA</li> </ul>
TM_NorthA merica	North America ACG	Global_Local_Ca talog	Target_Market_Hie rarchy	<ul style="list-style-type: none"> <li>• US</li> <li>• CA</li> </ul>



---

## Notices

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

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

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

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

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

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

---

## Programming interface information

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

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

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

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

---

## Trademarks and service marks

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

IBM  
the IBM logo  
AIX  
CrossWorlds  
DB2  
DB2 Universal Database  
Domino  
Lotus  
Lotus Notes  
MQIntegrator  
MQSeries  
Tivoli



## WebSphere

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

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

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

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

IBM 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

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.

