**LINK Cartridge for MainStreet Demandware Integration**

Version 1.0.0



Table of Contents

[1. Summary 3](#_Toc351566673)

[2. Component Overview 4](#_Toc351566674)

[2.1 General Functional Overview 4](#_Toc351566675)

[2.1.1 File Transfer Server 4](#_Toc351566676)

[2.1.2 File Transfer Handshake 5](#_Toc351566677)

[2.1.3 Error Recovery 5](#_Toc351566678)

[2.1.4 File Format 6](#_Toc351566679)

[2.1.5 Limitations, Constraints 6](#_Toc351566680)

[2.1.5.1 File Sizes 6](#_Toc351566681)

[2.1.6 Import into Production 6](#_Toc351566682)

[2.1.7 Compatibility 7](#_Toc351566683)

[3. Implementation Guide 8](#_Toc351566684)

[3.1 Cartridges 8](#_Toc351566685)

[3.1.1 Configuration 8](#_Toc351566686)

[3.1.2 Add Custom Cartridges to your Project Code Base 8](#_Toc351566687)

[3.1.3 Assign Cartridge to Sites 8](#_Toc351566688)

[3.2 Create Job Schedules 9](#_Toc351566689)

[3.3 Cartridge Structure:- 11](#_Toc351566690)

[3.3.1 Custom Pipelines:- 12](#_Toc351566691)

[3.3.2 Custom Scripts:- 12](#_Toc351566692)

[3.3.3 Templates 13](#_Toc351566693)

[3.3.4 Web References 13](#_Toc351566694)

[3.3.5 Custom Code 14](#_Toc351566695)

[3.4 Installing MainStreet Cartridge for Demandware 15](#_Toc351566696)

[3.5 Typical Project Plan 16](#_Toc351566697)

[3.5.1 Roles, Responsibilities 16](#_Toc351566698)

[3.5.2 Typical Efforts and Timelines 16](#_Toc351566699)

[3.6 DW-MainStreet Integration Cartridge Site Preferences 17](#_Toc351566700)

[4. Test Cases 17](#_Toc351566701)

[4.1 Scope of Testing:- 17](#_Toc351566702)

[4.1.1 Export Type Jobs:- 17](#_Toc351566703)

[4.1.2 Import Type Jobs:- 17](#_Toc351566704)

[5. Known Issues 19](#_Toc351566705)

[6. Release History 19](#_Toc351566706)

Summary

MainStreet cartridge for Demandware is used to exchange data with MainStreet Commerce’s Order Management System Businessflow.

The cartridge can be configured to Export or Import XML data feeds between the Demandware and Mainstreet Commerce’s Businessflow

All the feeds are XML files based on the Demandware import/export schema.

This cartridge supports the following Imports / Exports:

***Imports:-***

1. Import Catalogs – This imports categories and products from Mainstreet Commerce’s Businessflow.
2. Import PriceBooks – This imports pricing information from Mainstreet Commerce’s Businessflow.
3. Import Inventory Lists – This imports the inventory information from Mainstreet Commerce’s Businessflow.

***Exports:-***

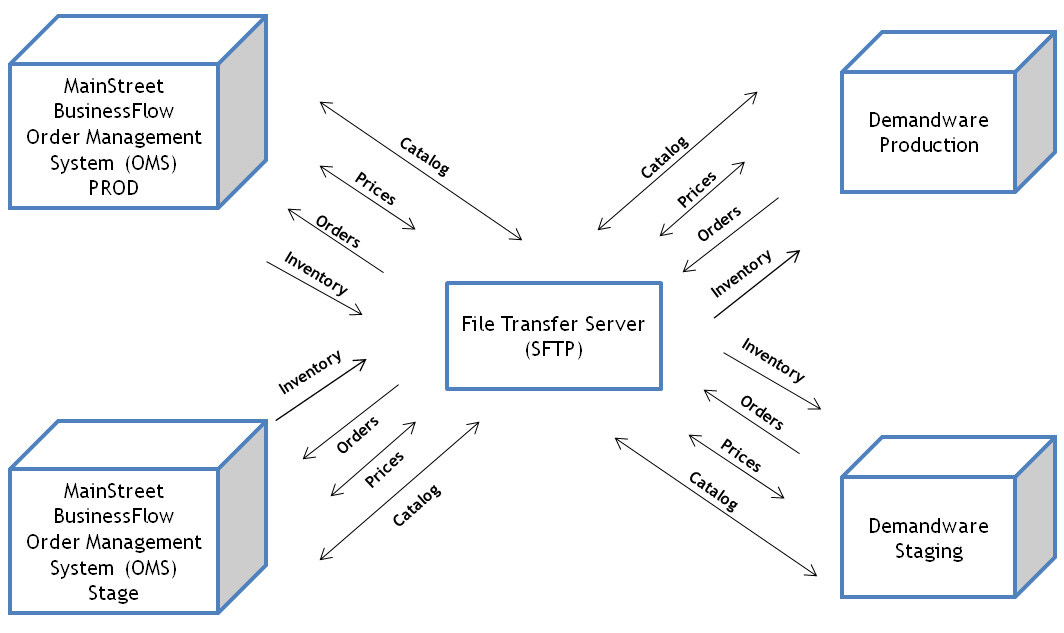
1. Export Catalogs – Exports categories and products under the given catalog to Mainstreet Commerce’s Businessflow.
2. Export PriceBooks – Exports pricing information to Mainstreet Commerce’s Businessflow.
3. Export Orders – Exports orders to Mainstreet Commerce’s Businessflow.

The remote SFTP server will be hosted by the MainStreet for exchanging the files.

The creation of the import files and consumption of the export files including the transformation into another format is outside the scope of this component. It will be done by the Demandware connector implemented at the MainStreet’s end. The connector has series of processes to exchange data with Demandware.

Component Overview

## General Functional Overview



### File Transfer Server

Idea of MainStreet Cartridge for Demandware Integration is to couple Demandware and MainStreet’s BusinessFlow (Order Management System) loosely by exchanging files on a Secure File Transfer Server. MainStreet Cartridge supports the protocol SFTP for the transfer with the File Transfer Server.

The Secure File Transfer Server will be hosted by MainStreet since Demandware does not offer hosting the Secure File Transfer Servers.

MainStreet will be creating the folders in the following structure in the Secure File Transfer server.

|  |  |  |
| --- | --- | --- |
| ***Development***  *businessflow*  *from-demandware*  *to-demandware* | ***Stage***  *businessflow*  *from-demandware*  *to-demandware* | ***Production***  *businessflow*  *from-demandware*  *to-demandware* |

MainStreet will give access privileges to these folders so that Demandware can exchange files through these folders for each of its instance (Development, Staging and Production).

### File Transfer Handshake

For incoming feeds MainStreet-DW Cartridge regularly lists the contents of a folder on the File Transfer Server and matches the file names against a pattern. All matching files (e.g. regular expression “^Catalog\_.MainStreet” will embed a timestamp in “yyyy-MM-dd\_HH-mm-ss” format using GMT time zone into \*\.xml$” are considered new data to be downloaded and imported. If there are multiple matching files, the file names are sorted alphabetically and processed in ascending order [e.g. “Catalog\_2009-03-26\_17-16-14.xml”]. This ensures uniqueness of file names in chronological processing order, even when switching to/from daylight saving time, and helps to correlate with log messages that typically use GMT time zone.

During upload of files to the File Transfer Server a temporary file name will be used. This is to ensure MainStreet Cartridge for Demandware won’t start downloading a file that hasn’t been uploaded completely. It is recommended to use the suffix “.tmp”, e.g. “Catalog\_2009-03-26\_17-16-14.xml.tmp”. Once the upload finishes successfully the client renames the file to its final name, e.g. “Catalog\_2009-03-26\_17-16-14.xml”. The configured matching rule (e.g. “^Catalog\_.\*\.xml$”) makes sure temporary files are not downloaded.

After successful download and import with either success or with permanent errors the file will be deleted at the File Transfer Server. This is to make sure the file won’t be processed again. The deletion of the file can also be used as an indicator by monitoring software to ensure the integration process completed processing the file.

Please note that even if the same feed content could be consumed by two Demandware instances (e.g. nightly price update), two files (one for each instance) needs to be uploaded to the File Transfer Server.

For outgoing feed like “Order Export” MainStreet-DW cartridge generates file names with embedded GMT time stamp and follows the conventions outlined above for upload and renaming the file.

### Error Recovery

Feed imports and exports are designed to recover automatically after a system failure (e.g. instance was shut down or crashed). After the failure the same process needs to be rerun.

After a successful download and import process the import file is deleted at the File Transfer Server as the very last step. If the process aborted due to system failure the process needs to be rerun and the file will be downloaded and imported again. All Demandware imports are idempotent (the same operation can be applied multiple times leading to the same result).

After successful order export, the orders are marked with export status EXPORTED. If the export process, file upload, or marking of the orders as exported fails due to system failure, the process needs to be rerun and an export file containing the previous orders will be created. There is a small chance for failure between successful file upload and marking the orders as exported. In that event the Demandware system will believe the orders haven’t been exported yet and they will be part of the next export. MainStreet will recognize repeated orders (i.e. subsequent orders with the same Demandware order ID) and will ignore subsequent transmission that way the orders are not duplicated in MainStreet BusinessFlow.

### File Format

The file format for the incoming and outgoing feeds is the standard Demandware import/export format. For XML files the schema definitions (XSD files) can be downloaded from Customer Central. A good approach to create a sample file is to set up the data as desired in Business Manager, run an export from the Business Manager UI to use the exported file as template.

All import files can alternatively be provided in gzip format (not to be confused with zip). In that case provide the file with extensions .xml.gz or .csv.gz at the File Transfer Server and adjust the file matching rule. The validation and import processes will automatically unzip the files during processing.

### Limitations, Constraints

### File Sizes

MainStreet DW Cartridge uses the class SFTP Client to download files. This client restricts single file downloads to 200 MB. If your import file is larger than 200 MB consider compressing it with gzip, otherwise the feed needs to be split into multiple files.

### Import into Production

Technically all object types can be imported on all Demandware instance types. Importing data directly into instance Production may have adverse effects as there is no preview or rollback functionality available. Please load all feeds that can be previewed (e.g. catalog data) into instance Staging for approval and use Demandware’s data replication feature for promotion to Production. Data replication also comes with rollback functionality.

When considering imports directly into Production mind that Demandware imports typically commit changes to the database on a per object basis. This means during the execution of the import some updates may already be visible in the storefront, others not. The catalog import is a two pass import: during the first pass objects are updated, during the second pass relationships between objects are updated. It is not recommended running catalog import directly into Production with MainStreet Cartridge for Demandware Integration.

### Compatibility

MainStreet Cartridge for Demandware Integration requires Demandware 2.10.0 or later.

Implementation Guide

## Cartridges

MainStreet-DW cartridge for data exchange between Demandware Instance and MainStreet BusinessFlow Instance. Please add that to your project code base.

**Cartridge Name: *int\_mainstreet***

This cartridge goes with the implementation of integration logic of different ImpExp (Import / Export) jobs. This needs to be assigned to any Storefront site and the Business Manager site. Modification of pipeline Custom\_FeedJobs is necessary to use additional feed configurations. Do not modify other code meant for standard imports / exports depicted in this document.

MainStreet Cartridge for DW Integration is configured with a FeedJobConfiguration Object. The standards Demandware Job Framework is used for monitoring and for scheduling the job execution of various Import / Export processes.

### Configuration

**Install and Configure MainStreet-DW Job Framework**

Since this implementation of MainStreet Cartridge for DW is based on the Demandware Job Framework, you need to enable this cartridge on your instance before you can proceed with the *int\_mainstreet* Cartridge.

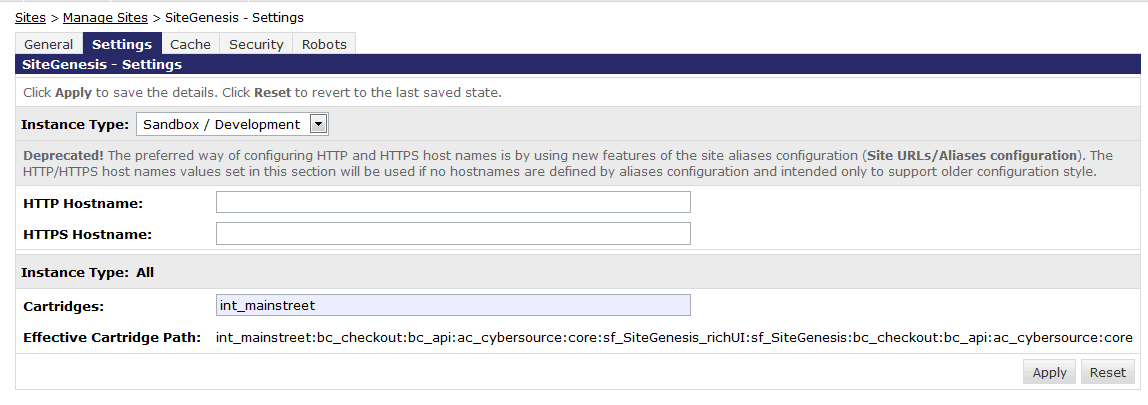
### Add Custom Cartridges to your Project Code Base

Add *int\_mainstreet* cartridge to your code repository (e.g. GIT) and import into Demandware Studio for upload to the server.

### Assign Cartridge to Sites

The feeds will be processed by Demandware jobs that are executed in the context of a site or organization. Pipelines for jobs are looked up in the cartridges at the Business Manager site, i.e. Sites-Site; templates (used for emails) are looked up in the cartridges for the respective storefront site.

In Business Manager go to Administration > Sites > Manage Sites and then to Sites-Site, and to each storefront site for which this integration is required. Under Cartridges add *int\_mainstreet*. Screen shot below.



## Create Job Schedules

Technically an arbitrary number of schedules and feed configurations can be set up in a Demandware instance.

For the MainStreet Cartridge for Demandware Implementation, those job configurations are placed in the Custom Object of type FeedJobConfiguration and the Pipeline Start Node will be specified in job schedule. Following are the steps to perform the job configuration setup in the given Demandware Instance

* Custom Object Type “*FeedJobConfiguration*” can be defined by importing the metadata file ***LINK\_mainstreet\metadata\CustomObjectTypes.xml.*** In Business Manager, the file ***CustomObjectTypes.xml*** needs to be imported under ***Administration 🡪 Site Development 🡪 Import & Export***
* All FeedJobConfiguration custom object records can be setup by importing the metadata file ***LINK\_mainstreet\metadata\FeedJobConfiguration.xml. In Business Manager, the file FeedJobConfiguration.xml*** needs to be imported under***Sites 🡪 Site-<SiteName> 🡪 Custom Objects 🡪 Import & Export***

The SFTP credentials, email ids and other settings in the FeedJobConfiguration.xml should be updated to reflect your configuration information before importing ***FeedJobConfiguration.xml*** into Business Manager.

* The job schedules that need to be setup for MainStreet Cartridge for Demandware Integration can be set up by importing the metadata file ***LINK\_mainstreet\metadata\JobSchedules.xml*. In Business Manager**, the above file needs to be imported under **Administration 🡪 Operations 🡪 Job Schedules 🡪 Import & Export**

Before importing the ***JobSchedules.xml, c***hange the **site-id**, **email-to** to set it up to run the job against the site you want and the email to which the job notification needs to be sent

* **For setting up the MainStreet preference group preferences (custom preferences), we can use the metadata file *LINK\_mainstreet\metadata\*** ***MainStreet-metadata.xml.* In Business Manager**, the above file needs to be imported under ***Administration 🡪 Site Development 🡪 Import & Export***

After setting up these configurations using the provided metadata files, you can edit the configuration values as per your requirements. Below would be the list of resulting jobs setup through the aforesaid metadata files.

|  |  |  |  |
| --- | --- | --- | --- |
| **Job Name** | **Description** | **Pipeline Name** | **Startnode** |
| ExportCatalogs | A job for Exporting the catalogs to MainStreet | Custom\_FeedJobs | StartExportCatalog |
| ExportOrders | A JOB for Exporting the orders to MainStreet | Custom\_FeedJobs | StartExportOrders |
| ExportPriceBooks | Exports the List Prices of the products in the catalogs | Custom\_FeedJobs | StartExportPriceBooks |
| ImportCatalog | Import complete categories of the catalog into Demandware | Custom\_FeedJobs | StartImportCatalog |
| ImportInventory | Imports the inventory from MainStreet BusinessFlow into Demandware | Custom\_FeedJobs | StartImportInventoryLists |
| ImportPriceBooks | Imports the Price Books from MainStreet BusinessFlow | Custom\_FeedJobs | StartImportPriceBooks |

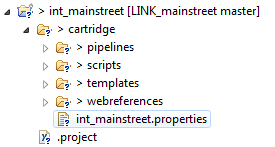
These jobs are invoked through job pipelines that contain one or more workflows / business flows each of which begin with the start node and terminate with an interaction node, interaction continue node or end node. These job pipelines are developed custom and typically perform the following

* **Export Job Pipeline** 🡪 Invokes the export process through DW API (this comes with the DW cartridge setup) and generates the files. Then pushes the exported files to the external system through SFTP.
* **Import Job Pipeline** 🡪 Invokes the file pull from the external system using the SFTP, downloads the files into the local folder (in the DW cloud), then feeds those files to the import process and invokes them. After finishing the import, archives and deletes the files from the remote location to avoid duplicate processing.

Pipelines and their underlying scripts, Pipelets and templates are bundled together into a single unit called Cartridge. Cartridges typically have the custom pipelines that automate the DW processes or integrate DW with external systems.

## Cartridge Structure:-

Following is the high level structure of the MainStreet Demandware Cartridge:

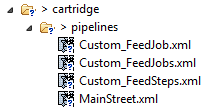


Here,

* ***int\_mainstreet*** 🡪 Cartridge Folder Name (Name specifies that this is the cartridge used for integrating with MainStreet BusinessFlow Order Management System)
* ***cartridge*** 🡪 Folder containing the source code files that performs various functions
* ***pipelines*** 🡪 Subfolder under cartridge folder. This contains the pipelines (High Level self-contained components that integrate with Demandware instance of the customer)
* ***scripts*** 🡪 Subfolder under cartridge folder. This contains the scripts to execute the high level functions within the pipelines (called pipelets). These scripts have the extension .ds
* ***templates*** 🡪 These are the ISML templates that can define the rendering / functional behavior of the Demandware storefront website. These are used to customize the storefront website.
* ***webreferences*** 🡪 If we need to have Demandware communicate with external systems for getting information it requires in store front website, we need to expose those information in the form of webServices and we can put those web service references under this cartridge folder so DW storefront website can make use of it.

Following are the brief descriptions of each of the above components:-

### Custom Pipelines:-



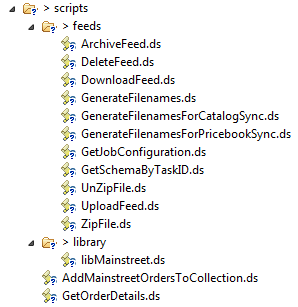
*Custom\_FeedJobs* is the parent pipeline that starts the execution of the job. This pipeline contains the start nodes for different jobs. For Example, StartExportCatalog

*Custom\_FeedJob* Pipeline gets the control from Custom\_FeedJobs. It acts as the controller and controls the job execution by directing to the appropriate step in the Custom\_FeedSteps pipeline.

*Custom\_FeedSteps* pipeline calls the pipelet scripts to perform the job execution. All the job configurations, custom objects and attributes are available to these pipelines. Hence those are read and initialized into the Pipeline Global variables.

*Orders pipeline* is the pipeline to demonstrate how to pull the data from MainStreet BusinessFlow using the webServices exposed by MainStreet

### Custom Scripts:-



Custom scripts are used to perform the actual tasks underlying the custom pipelines directed by the pipelets inside those pipelines. Above are some of the custom scripts. Here are the brief descriptions of some of them.

*GetJobConfiguration.ds* script gets the job configuration information defined in the Custom Object taskXML attribute and loads that into the Global pipeline variables.

*UploadFeed.ds* script uploads the exported files (from the export process) into the remote SFTP locations

*DownloadFeed.ds* script downloads the files from the remote SFTP locations and makes it available for import into Demandware.

*DeleteFeed.ds* script deletes the feed file from the remote location once the processing on the file is done. This avoids the duplicate execution of the process on those files.

*ArchiveFeed.ds* archives the files in the local DW directory

*GetFileNames.ds* returns the file name of the feeds adhering to the convention

These scripts have the main method called “execute” in each of them that gets invoked when the scripts are triggered from the Pipelines.

*library/libMainStreet.ds* has the logic to connect to the MainStreet WebServices, instantiate the service object to access the webServices methods. Also contains the logic to make calls to specific methods like GetListOrdersByOrderId, GetListOrdersByShipOrderId, GetListOrdersByCustomer etc.

The other scripts under the scripts folder like AddMainstreetOrdersToCollection.ds, GetOrderDetails.ds can be used by the pipelines that interact with the Storefront websites; to get the MainStreet BusinessFlow Order Status to Demandware storefront.

### Templates

Contains templates used by unit test pipeline for the web services. Also contains custom email templates used in emails sent by Demandware Job Framework

### Web References



Demandware may need to get information from external systems like MainStreet (OMS) on Order Status, Customer Details etc., so that they can be used in displaying in the storefront websites or in custom processes. To accomplish this, the webServices exposed from those external systems would be referenced from within the DW cartridge and are accessed through the Demandware scripts. Demandware has this ability to pull real time data from external systems to keep their customers up to date on the status of their orders or any queries placed to the merchant.

The end points of these web Services are configured in the Meta data XML (MainStreet-metadata.xml)

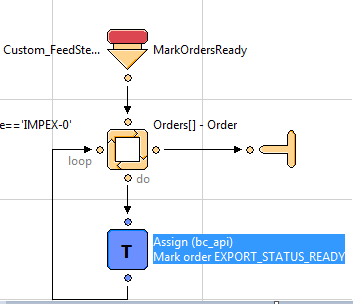
The authentication information to access the MainStreet WebServices are stored in the custom preferences screen under the path “Custom Site Preference Groups > MainStreet”

### Custom Code

Order export in MainStreet Cartridge for DW Integration exports all orders that come with export status READY and “exportAfter” attribute (a Demandware system may be configured not to export orders for a configured period of time) has been exceeded. The reference application SiteGenesis currently does not set the export status to READY. So the order creation pipeline is modified in the project code to set dw.order.Order.exportStatus to dw.order.Order.EXPORT\_STATUS\_READY.

**Steps to set the Order Export Status to “READY”**

1. Below pipeline “Mark order EXPORT\_STATUS\_READY” will be created and called as part of Order Creation Pipeline. In this, an assign node is added as indicated in the below diagram.



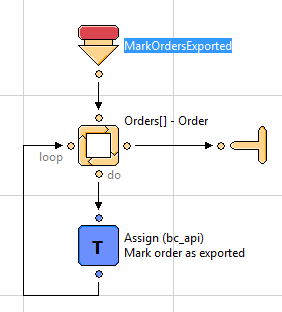
1. From\_0 node value is set to “dw.order.Order.EXPORT\_STATUS\_READY” as below in the code for above pipeline (Custom\_FeedSteps.xml)

<key-binding alias="dw.order.Order.EXPORT\_STATUS\_READY" key="From\_0"/>

1. To\_0 node value is set to “Order.exportStatus” as below in the code for above pipeline.

<key-binding alias="Order.exportStatus" key="To\_0"/>

After the orders are exported and the order file is pushed to remote SFTP location successfully, we need to mark those orders exported in the system so they will not be picked up by the Order Export job again. A custom pipeline “MarkOrdersExported” from this cartridge will be used to do that as defined in the below diagram



**Steps to set the Order Export Status to “EXPORTED”**

1. Above pipeline “MarkOrdersExported” will be created and called as part of Order Creation Pipeline. In this, an assign node is added as indicated in the above diagram.
2. From\_0 node value is set to “dw.order.Order.EXPORT\_STATUS\_EXPORTED” as below in the code for above pipeline ((Custom\_FeedSteps.xml)

<key-binding alias="dw.order.Order.EXPORT\_STATUS\_EXPORTED" key="From\_0"/>

1. To\_0 node value is set to “Order.exportStatus” as below in the code for above pipeline.

<key-binding alias="Order.exportStatus" key="To\_0"/>

## Installing MainStreet Cartridge for Demandware

DW Cartridges are installed using Demandware UX Studio which is a development plugin for Eclipse IDE. Once the cartridges are setup in the UX Studio from the local drive, they will be uploaded and installed against the given Demandware Server instance (Development / Staging / Production).

The cartridge will be connected to the D/W server using the DW Server Connection Project.

Once the cartridge and the DW server instance are linked together, any changes to the cartridge files from the UX studio will be reflecting in the DW server source and the effect can be seen immediately.

UX Studio also helps in debugging DW Cartridge files (Pipeline and Script Debugging) using the “Debug Perspective” option.

## Typical Project Plan

### Roles, Responsibilities

Typically most of the integration work is done by the backend developer. We expect that the person doing this integration is familiar with the web service, xml processing and has hands on experience with the Demandware platform.

### Typical Efforts and Timelines

|  |  |  |
| --- | --- | --- |
| **LINK Cartridge for Mainstreet DW Integ** | **Level of Effort (LOE)** | **Dependencies** |
| Initial Cartridge Setup | **0.5** - Person Day  List of tasks involved:   * Add DW-MainStreet Integration Cartridge to the project * Setup the Demandware Server Connection Project against the given Demandware instance | * Cartridge is available |
| Configure Export / Import jobs as requested by the client | **0.5** - Person Day  List of tasks involved:   * Create job configuration XMLs for different jobs and set them up in the FeedJobConfiguration attributes * Setup the jobs associating them to the respective pipelines and start nodes, configure the notification, resources, parameters and next actions to success / failure scenarios |  |
| Testing exchange of files by running export and import jobs. | **0.5** - Person Day  List of tasks involved:   * Run the jobs one by one to make sure they work fine in terms of transferring the data, deleting / archiving the files | * Coordinate with the person from MainStreet to make sure their system is setup to generate, upload and download, consume the files from Demandware * Ensure data gets updated properly between the two systems |
| Unit Testing MainStreet Web Services | **0.1** - Person Day  List of tasks involved:   * Use the unit test scripts from the cartridge and test the web Services to make sure they return the data as expected from MainStreet | * Coordinate with MainStreet for any clarification required with the web service and its methods. |

Typically most of the integration work is done by the backend developer. We expect that the person doing this integration is familiar with the web service, xml processing and has hands on experience with the Demandware platform.

## DW-MainStreet Integration Cartridge Site Preferences

|  |  |
| --- | --- |
| **Site Preferences** | **Description** |
| MainStreetDomain | MainStreet BusinessFlow Domain Name |
| MainStreetUsername | Username of the BusinessFlow service account |
| MainStreetPassword | Password for the BusinessFlow service account |
| MainStreetEndpoint | End point of the BusinessFlow Order Status WebService |

# Test Cases

## Scope of Testing:-

### Export Type Jobs:-

These jobs trigger the Demandware Export API to generate the outbound files and push them into the SFTP location under “from-demandware” folder. It should be good enough to run these jobs and see if they generate and drop the valid files in the common SFTP folder.

### Import Type Jobs:-

These jobs pull inbound files from remote common SFTP folder and import data from those files into Demandware using Demandware Import API. It should be good enough to run these jobs and see if the files are pulled from remote SFTP, cleared in remote SFTP location and the data from those files are imported into Demandware.

Following are few basic instructions to test these import / export jobs:-

|  |  |
| --- | --- |
| **TC #1** | TC/MS/DW/CAR/EXPCAT |
| **Test Case** | Testing Export Catalog Job |
| **Description** | 1. In Demandware Business Manager, go to Administration 🡪 Operations 🡪 Job Schedules 🡪 Export Catalog.  2. Run Export Catalog job by clicking the “Run” button at the bottom left.  3. The job status will be shown as a new job entry at the bottom.  4. Wait for the job to complete. Keep refreshing using the “Refresh” button at the bottom left. |
| **Expected Results** | 1. After the job completes successfully, a newly generated Catalog file should appear under “from-demandware” folder of remote SFTP location (configured for ExportCatalog job)  2. Upon checking few product records from the file, they should match the data exactly as in Demandware Instance from where the job was run. |

|  |  |
| --- | --- |
| **TC #2** | **TC/MS/DW/CAR/EXPPB** |
| **Test Case** | **Testing Export PriceBooks Job** |
| **Description** | **1. In Demandware Business Manager, go to Administration 🡪 Operations 🡪 Job Schedules 🡪 Export PriceBooks.**  **2. Run Export PriceBooks job by clicking the “Run” button at the bottom left.**  **3. The job status will be shown as a new job entry at the bottom.**  **4. Wait for the job to complete. Keep refreshing using the “Refresh” button at the bottom left.** |
| **Expected Results** | **1. After the job completes successfully, a newly generated PriceBooks file should appear under “from-demandware” folder of remote SFTP location (configured for ExportPriceBooks job)**  **2. Upon checking few PriceBooks records from the file, they should match the data exactly as in Demandware Instance from where the job was run.** |

|  |  |
| --- | --- |
| **TC #3** | **TC/MS/DW/CAR/EXPORD** |
| **Test Case** | **Testing Export Orders** |
| **Description** | **1. Make sure all the orders to be exported from Demandware are marked with the export status “Ready for Export” and confirmation status “Confirmed”. If none of them are “Ready for Export”, no order will get exported**  **2. In Demandware Business Manager, go to Administration 🡪 Operations 🡪 Job Schedules 🡪 Export Orders**  **3. Run Export Orders job by clicking the “Run” button at the bottom left.**  **4. The job status will be shown as a new job entry at the bottom**  **5. Wait for the job to complete. Keep refreshing using the “Refresh” button at the bottom left.**  **6. After checking the results, repeat the steps 1 to 5 for few new orders by changing the order status to “Ready for Export” to see if the job exports those new orders.** |
| **Expected Results** | **1. After the job completes successfully, a newly generated Order file should appear under “from-demandware” folder of remote SFTP location (configured for ExportOrders job)**  **2. Upon checking few Order records from the file, they should match the data exactly as in Demandware Instance from where the job was run.**  **3. After the export job, the exported orders should be marked ‘Exported” in the Demandware Instance. You can check this under Sites-<SiteName> 🡪 Ordering 🡪 Search Order 🡪 Order Detail** |

|  |  |
| --- | --- |
| **TC #4** | **TC/MS/DW/CAR/IMPCAT** |
| **Test Case** | **Testing Import Catalog Job** |
| **Description** | **1. Make sure the Catalog file is ready to be imported in the remote SFTP location (Folder Name: to-demandware) from MainStreet BusinessFlow. If not, contact MainStreet to get the file in that location. Make a copy of the file for further testing.**  **2. In Demandware Business Manager, go to Administration 🡪 Operations 🡪 Job Schedules 🡪 Import Catalog**  **3. Run Import Catalog job by clicking the “Run” button at the bottom left.**  **4. The job status will be shown as a new job entry at the bottom.**  **5. Wait for the job to complete. Keep refreshing using the “Refresh” button at the bottom left.**  **6. After checking the results, change the product data in the file (took from the above copy), upload the file to remote SFTP and repeat the steps 1 to 5 to see if the job imports the product data changes into Demandware** |
| **Expected Results** | **1. After the job completes successfully, the file should have been deleted from the remote SFTP location**  **2. Upon checking few product records in Demandware instance, they should match the data exactly as in the imported file.** |

|  |  |
| --- | --- |
| **TC #5** | **TC/MS/DW/CAR/IMPPB** |
| **Test Case** | **Testing Import PriceBooks Job** |
| **Description** | **1. Make sure the PriceBooks file is ready to be imported in the remote SFTP location (Folder Name: to-demandware) from MainStreet BusinessFlow. If not, contact MainStreet to get the file in that location. Make a copy of the file for further testing.**  **2. In Demandware Business Manager, go to Administration 🡪 Operations 🡪 Job Schedules 🡪 Import PriceBooks**  **3. Run Import PriceBooks job by clicking the “Run” button at the bottom left.**  **4. The job status will be shown as a new job entry at the bottom.**  **5. Wait for the job to complete. Keep refreshing using the “Refresh” button at the bottom left.**  **6. After checking the results, change the product price data in the file (took from the above copy), upload the file to remote SFTP and repeat the steps 1 to 5 to see if the job imports the price changes into Demandware** |
| **Expected Results** | **1. After the job completes successfully, the file should have been deleted from the remote SFTP location**  **2. Upon checking few product pricing records in Demandware instance, they should match the data exactly as in the imported file.** |

|  |  |
| --- | --- |
| **TC #6** | **TC/MS/DW/CAR/IMPINV** |
| **Test Case** | **Testing Import Inventory Job** |
| **Description** | **1. Make sure the Inventory file is ready to be imported in the remote SFTP location (Folder Name: to-demandware) from MainStreet BusinessFlow. If not, contact MainStreet to get the file in that location. Make a copy of the file for further testing.**  **2. In Demandware Business Manager, go to Administration 🡪 Operations 🡪 Job Schedules 🡪 Import Inventory**  **3. Run Import Inventory job by clicking the “Run” button at the bottom left**  **4. The job status will be shown as a new job entry at the bottom.**  **5. Wait for the job to complete. Keep refreshing using the “Refresh” button at the bottom left.**  **6. After checking the results, change the inventory data in the file (took from the above copy), upload the file to remote SFTP and repeat the steps 1 to 5 to see if the job imports the inventory changes into Demandware** |
| **Expected Results** | **1. After the job completes successfully, the file should have been deleted from the remote SFTP location**  **2. Upon checking few product inventory records in Demandware instance, they should match the data exactly as in the imported file.** |

Known Issues

None

# Release History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 1.0.0 | 02/21/2013 | Initial release |