IBM WebSphere Commerce Accelerator

Informatica MDM - Product 360

Table of Contents

1 Functional scope 8

2 Installation Guide 9

2.1 Pre-Installation Checklist 9

2.1.1 WCS environment 9

2.1.2 Product 360 environment 9

2.2 Install the IBM WebSphere Accelerator 9

2.2.1 Download the Product 360 - WebSphere Accelerator zip 9

2.2.2 Extract the WebSphere Accelerator zip 9

2.2.3 WCS environment configuration 10

2.2.3.1 Install Java 10

2.2.3.2 Install Informatica Funcd for WCS Accelerator 10

2.2.4 Product 360 environment configuration 15

2.2.4.1 Product 360 server 15

2.2.4.2 Product 360 client 15

3 Attribute Dictionary 15

3.1 What is an attribute dictionary? 15

3.2 How does the attribute dictionary model match the Product 360 attribute model? 15

3.2.1 The problem 15

3.2.2 The solution: One pool feature - many structure group attributes - some attributes for the attribute dictionary 16

3.2.3 Example: Cumulating attributes 16

3.2.4 Example: Collecting preset values of attributes 17

3.2.5 Limitation 18

3.2.5.1 Computing the list of dictionary attributes 18

3.2.5.2 Multi value support 18

3.2.5.3 Product 360 field - WCS field mapping 18

3.2.5.4 Deleted allowed values, deleted attributes 19

3.3 Attributes 19

3.3.1 Types of dictionary attributes 19

3.3.1.1 Allowed value attributes - assigned value attributes 19

3.3.1.2 Defining attributes - describing attributes 19

3.3.2 Export enhancements 19

3.4 Allowed values ... 20
3.4.1 Export enhancements ... 20
3.5 Attribute - catalog entry mapping ... 20
3.5.1 Export enhancements ... 21
3.5.1.1 "GetWCSAttributeUsage" export function ... 21
3.5.1.2 "Attribute dictionary attribute relations" sub-data type ... 21
3.6 New and changed attributes and values ... 21
4 Data Exchange ... 22
4.1 Scenarios ... 22
4.1.1 Limitations: ... 22
4.2 General Workflow ... 22
4.3 WCS Data Loader Environment ... 22
4.3.1 Overview ... 22
4.3.1.1 Files of the dataload directory ... 23
4.3.1.2 Files of the object-loaders.xml directory ... 24
4.3.2 WCS object loader mapping ... 24
4.3.2.1 Data load sequence of structure ... 24
4.3.2.2 Data load sequence of entries ... 25
4.3.2.3 Data load sequence of prices ... 26
4.4 Workflows ... 27
4.4.1 Full load ... 27
4.4.1.1 Export template ... 27
4.4.1.2 Exported data ... 28
4.4.1.3 Created Files ... 28
4.4.1.4 WCS loader ... 28
4.4.1.5 Detailed Overview of full export ... 29
4.4.2 Delta load ... 30
4.4.2.1 Export template ... 30
4.4.2.2 Exported data ... 30
4.4.2.3 Created Files ... 30
4.4.2.4 WCS loader ... 31
4.4.3 Immediate Update ... 31
4.4.3.1 Export template ... 32
4.4.3.2 Exported data ... 32

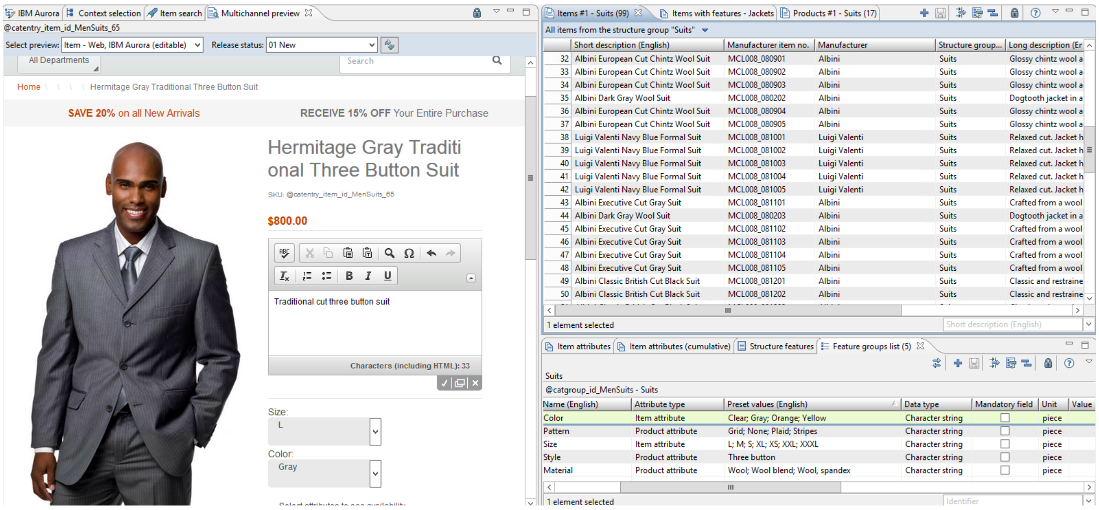
4.4.3.3 Created Files ... 32
4.4.3.4 WCS loader ... 32
4.4.4 Delete all data ... 33
4.4.4.1 Export template ... 34
4.4.4.2 Exported data ... 34
4.4.4.3 Created Files ... 34
4.4.4.4 WCS loader ... 35
4.5 WCS Data Load Configuration ... 35
4.5.1 The data load utility configuration files ... 35
4.5.1.1 The environment configuration file ... 35
4.5.1.2 The configuration files for the data load sequence ... 37
5 Data Maintenance ... 38
5.1 Objective ... 38
5.2 The catalog groups' data maintenance ... 39
5.2.1 Catalog groups maintenance in Product 360 ... 39
5.2.2 Data fields of structure groups used for the standard interface ... 42
5.3 The catalog entries' data maintenance ... 43
5.3.1 Catalog Entries Description ... 46
5.3.1.1 Language dependent texts ... 46
5.3.1.2 Images ... 46
5.3.2 Catalog Entries Prices ... 47
5.3.2.1 Prices in WebSphere Commerce ... 47
5.3.3 Attributes ... 48
5.3.3.1 Attributes in WebSphere Commerce ... 48
5.3.3.2 Attributes in Product 360 ... 49
5.4 Attribute Dictionary ... 52
5.4.1 Computing the list of dictionary attributes ... 53
5.4.2 Multi value support ... 53
5.4.3 Product 360 field - WCS field mapping ... 53
5.4.3.1 Features without preset values ... 53
5.4.3.2 Item/Product attributes ... 54
5.4.4 Deleted allowed values, deleted attributes ... 54
5.5 Exports ... 54
5.5.1 Limitations ... 56

5.5.1.1 Exporting into the master catalog of IBM WebSphere 56
5.5.1.2 Descriptive products attributes without preset values 56
5.5.1.3 Delta Export 56
5.5.1.4 Immediate Export 57
6 Field mapping 57
6.1 Overview 57
6.2 Formatting and mapping for export format templates 57
6.2.1 Numbers 57
6.2.2 Dates 58
6.2.3 Language 58
6.2.4 Data type 58
6.2.5 Association type 58
6.2.6 Sequence 59
6.2.7 Strings 59
6.2.8 Keywords 59
6.2.9 Images 59
6.3 Data consistency 60
6.3.1 Data field length 60
6.4 The Attribute Dictionary data exchange 60
6.4.1 Attribute Dictionary - Attributes 60
6.4.1.1 Template Configuration 61
6.4.1.2 Sample for an attribute dictionary attributes' import file 61
6.4.1.3 Field mapping 61
6.4.2 Attribute Dictionary – Allowed Values 65
6.4.2.1 Template Configuration 65
6.4.2.2 Sample for an 'attribute dictionary attributes allowed values' import file 66
6.4.2.3 Field mapping 66
6.5 The catalog groups data exchange 69
6.5.1 Catalog Groups 69
6.5.1.1 Template Configuration 69
6.5.1.2 Sample for a catalog groups' import file 70
6.5.1.3 Field mapping 70
6.5.2 Catalog Groups Description 71
6.5.2.1 Template Configuration 72

6.5.2.2 Collecting image data ... 72
6.5.2.3 Sample for a catalog groups' description import file ... 73
6.5.2.4 Field mapping ... 73
6.6 The catalog entries data exchange ... 75
6.6.1 Catalog Entries ... 75
6.6.1.1 Template Configuration ... 75
6.6.1.2 Sample for a catalog entries' import file ... 76
6.6.1.3 Field mapping ... 76
6.6.2 Catalog Entries Description ... 78
6.6.2.1 Template Configuration ... 78
6.6.2.2 Collecting image data ... 79
6.6.2.3 Sample for a catalog entries' descriptions import file ... 80
6.6.2.4 Field mapping ... 80
6.6.3 Catalog Entries' Attribute Values ... 83
6.6.3.1 Template Configuration ... 83
6.6.3.2 Sample for a catalog entries' attribute values import file ... 83
6.6.3.3 Field mapping ... 83
6.6.4 Catalog Entries' Associations ... 85
6.6.4.1 Template Configuration ... 85
6.6.4.2 Sample for a catalog entries' associations import file ... 86
6.6.4.3 Field mapping ... 86
6.6.5 Catalog Entries Prices - list prices ... 87
6.6.5.1 Template Configuration ... 87
6.6.5.2 Sample for a catalog entries' list price import file: ... 88
6.6.5.3 Field mapping ... 88
6.6.6 Catalog Entries Prices - offer prices ... 89
6.6.6.1 Template Configuration ... 89
6.6.6.2 Sample for a catalog entries' offer price import file ... 89
6.6.6.3 Field mapping ... 90
6.7 Appendix A: Variables of the Export Templates ... 91
6.8 Appendix B: Language Mapping ... 93
6.9 Appendix C: Data Types Mapping ... 94
6.10 Appendix D: Item Reference Types Mapping ... 94
6.11 Appendix E: Object Types Mapping ... 95

6.12 Repository configuration 95
6.12.1 Changes in Repository Types 95
6.12.2 Changes in repository custom 96
6.12.3 Added fields to Structure Value 97
6.12.3.1 Added fields to Structure Group 97
6.12.3.2 Added fields to Item 97
6.12.3.3 Added fields to Product 98
7 Sample data 99
7.1 How to add sample data 99

This document describes the data exchange interface between Informatica Product 360 Version 8.0.6 and IBM WebSphere Commerce Version 8.



IBM Aurora
Context selection
Item search
Multichannel preview

@catentry_item_id_MenSuits_65

Select preview: Item - Web, IBM Aurora (editable)   Release status: 01 New

All Departments
Search
Home
Hermitage Gray Traditional Three Button Suit
SAVE 20% on all New Arrivals
RECEIVE 15% OFF Your Entire Purchase

Hermitage Gray Traditional Three Button Suit

SKU: @catentry_item_id_MenSuits_65

$800.00

Traditional cut three button suit

Characters (including HTML): 33

Size: L

Color: Gray

Items #1 - Suits (99)
Items with features - Jackets
Products #1 - Suits (17)

All items from the structure group "Suits"

	Short description (English)	Manufacturer item no.	Manufacturer	Structure group...	Long description (Er
32	Albini European Cut Chintz Wool Suit	MCL008_080901	Albini	Suits	Glossy chintz wool a
33	Albini European Cut Chintz Wool Suit	MCL008_080902	Albini	Suits	Glossy chintz wool a
34	Albini European Cut Chintz Wool Suit	MCL008_080903	Albini	Suits	Glossy chintz wool a
35	Albini Dark Gray Wool Suit	MCL008_080202	Albini	Suits	Dogtooth jacket in a
36	Albini European Cut Chintz Wool Suit	MCL008_080904	Albini	Suits	Glossy chintz wool a
37	Albini European Cut Chintz Wool Suit	MCL008_080905	Albini	Suits	Glossy chintz wool a
38	Luigi Valenti Navy Blue Formal Suit	MCL008_081001	Luigi Valenti	Suits	Relaxed cut. Jacket h
39	Luigi Valenti Navy Blue Formal Suit	MCL008_081002	Luigi Valenti	Suits	Relaxed cut. Jacket h
40	Luigi Valenti Navy Blue Formal Suit	MCL008_081003	Luigi Valenti	Suits	Relaxed cut. Jacket h
41	Luigi Valenti Navy Blue Formal Suit	MCL008_081004	Luigi Valenti	Suits	Relaxed cut. Jacket h
42	Luigi Valenti Navy Blue Formal Suit	MCL008_081005	Luigi Valenti	Suits	Relaxed cut. Jacket h
43	Albini Executive Cut Gray Suit	MCL008_081101	Albini	Suits	Crafted from a wool
44	Albini Dark Gray Wool Suit	MCL008_080203	Albini	Suits	Dogtooth jacket in a
45	Albini Executive Cut Gray Suit	MCL008_081102	Albini	Suits	Crafted from a wool
46	Albini Executive Cut Gray Suit	MCL008_081103	Albini	Suits	Crafted from a wool
47	Albini Executive Cut Gray Suit	MCL008_081104	Albini	Suits	Crafted from a wool
48	Albini Executive Cut Gray Suit	MCL008_081105	Albini	Suits	Crafted from a wool
49	Albini Classic British Cut Black Suit	MCL008_081201	Albini	Suits	Classic and restraine
50	Albini Classic British Cut Black Suit	MCL008_081202	Albini	Suits	Classic and restraine

1 element selected

Suits

@catgroup_id_MenSuits - Suits

Name (English)	Attribute type	Preset values (English)	Data type	Mandatory field	Unit	Value
Color	Item attribute	Clear; Gray; Orange; Yellow	Character string	□	piece	
Pattern	Product attribute	Grid; None; Plaid; Stripes	Character string	□	piece	
Size	Item attribute	L; M; S; XL; XS; XXL; XXXL	Character string	□	piece	
Style	Product attribute	Three button	Character string	□	piece	
Material	Product attribute	Wool; Wool blend; Wool, spandex	Character string	□	piece	

1 element selected

The commerce accelerator supports the new IBM attribute dictionary model that enables reduced maintenance and reuse across multiple products. When a common attribute in the dictionary is changed it is reflected in the attribute for all products in the catalog thereby simplifying the process of keeping attribute names and values consistent across a commerce site.

It also enables a commerce site to provide complex search, filter and product comparison capabilities which helps provide enhanced service to the customer.

The accelerator can be used to simplify and expedite the implementation of common use cases by providing support for hierarchies, descriptions, media assets pricing information, marketing associations and attributes out of the box.

The built in data quality module and the preview templates of the Product 360 application ensures the quality, envisioning of the product on how it would display on the commerce site to take the appropriate action and conformity of the data which is transferred to the commerce site.

Different export scenarios pertaining to full catalog, delta or the possibility to trigger the publication of selected products instantly with just a click are supported.

Overall it helps improve operations by providing sophisticated capabilities to manage products and catalogs to deliver a seamless, cross-channel shopping experience for commerce.

To verify the data quality of the assortment which will be published to the IBM store the data quality engine of the Product 360 system should to be used. This ensure that no errors occur during the load of the IBM commerce system.

Please consider the allowed lengths of the IBM Websphere Commerce system which are documented on the field mappings site.

1 Functional scope

The accelerator features three ways to transfer data from the Product 360 System to the IBM WebSphere Commerce System.

The full data load is designed to transfer all relevant information, including the attribute dictionary, catalog groups, product and item information to IBM, often used for an initial load of the data.
The delta data load is designed to run scheduled in an recurrent manner and transfer the changes which happend since the last run to the IBM system. It takes not only product and item changes but also changes in the attribute dictionary or taxonomy into account. Changing the parent of a catalog group is not supported, as well as changing the type or target of a marketing association of an item or a product.
The third way to transfer data to the IBM system is the immediate export. With this functionality it is possible to transfer a single item or product, as well as a selection of items to the IBM store. This functionality is designed to work for smaller time-critical changes of an item, like changing text information and updating it as soon as possible in the store. Structural changes (like changing the attribute dictionary, catalog groups or group assignment) are not in the scope of this data load.

2 Installation Guide

The Product 360 WebSphere Commerce Accelerator will be available as ZIP file. Specific folders of this ZIP have to get moved to the Product 360 environment, others to the IBM WebSphere Commerce Shop environment.

2.1 Pre-Installation Checklist

2.1.1 WCS environment

An IBM WebSphere Commerce 8 must be installed.

A file share must be available for the Product 360 export and the dataload into the IBM WebSphere Commerce Shop server (**<SHARED_DIRECTORY_PATH>**). We assume that this is `D:/Filetransfer`. In case you chose any other directory, you need to change some paths which will be described later.

2.1.2 Product 360 environment

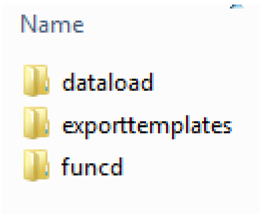
The version of the installed Product 360 must be 8.0.6 or higher.

2.2 Install the IBM WebSphere Accelerator

2.2.1 Download the Product 360 - WebSphere Accelerator zip

2.2.2 Extract the WebSphere Accelerator zip

Unpack the **Product 360_<*Version*>_<Revision>_resources_websphere.delta.zip** to a temporary directory on the Product 360 environment.



Name

dataload

exporttemplates

funcd

The temporary directory is containing files for the Informatica Product 360 environment and also for the IBM WebSphere Commerce environment.

2.2.3 WCS environment configuration

Copy following files from the unpacked **Product 360_<*Version*>_<Revision>_resources_websphere.delta.zip** to the **<SHARED_DIRECTORY_PATH>** on the IBM WebSphere Commerce environment:

- dataload
- funcd

Open the `wc-dataload-env.xml` in the `dataload` folder and adjust the database configuration.

If your **<SHARED_DIRECTORY_PATH>** differs to the standard path (`D:/Filetransfer`) the following files in the `dataload` directory have to get adjusted:

- buildSOLRindex.bat
- buildSOLRindex_Delta.bat
- cleanupFolders.bat
- load_multimedia.bat

If the installation path of the IBM WebSphere Commerce differs to `D:\IBM\WCDE_ENT70` and the catalog id is not `10001`, please adjust following files in the `dataload` directory:

- load_multimedia.bat
- load1-structure.bat
- load2-entries.bat
- load2-entries.index.bat
- load3-prices.bat
- loadZ-delete.bat

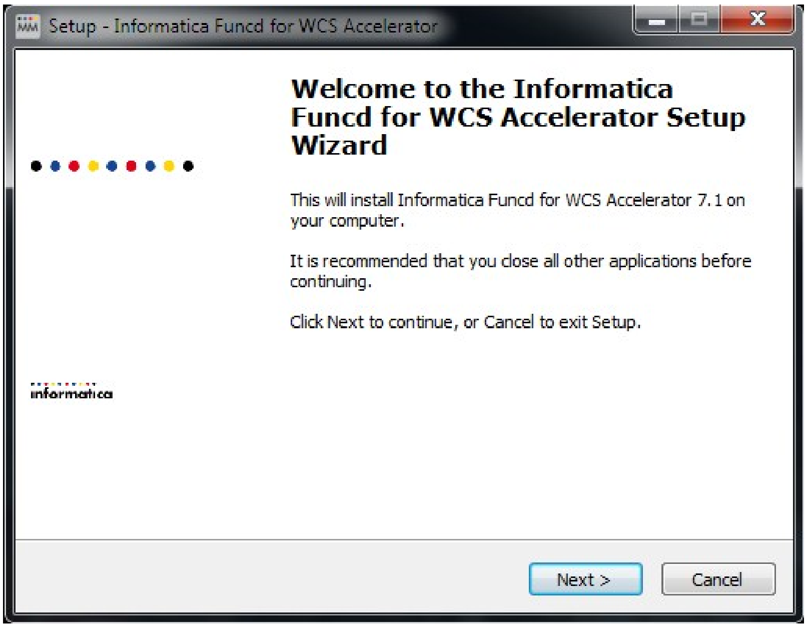
2.2.3.1 Install Java

The images will be transfered in zips from Product 360 to WCS. For unzipping those, Java JDK is needed. If you took another path than `C:\Program Files (x86)\Java\jdk1.7.0_25`, please adjust the `dataload\load_multimedia.bat` file.

2.2.3.2 Install Informatica Funcd for WCS Accelerator

This tool is needed, to poll a specific directory. The Product 360 export will put files inside, which will trigger the necessary bat or cmd file to start the data load in the IBM WebShere Commerce.

1. Unpack the zip inside the `funcd` folder in the share folder of the IBM WebSphere Commerce environment
2. Execute the `Setup_WCS_Funcd.exe`
3. Click Next



Setup - Informatica Funcd for WCS Accelerator

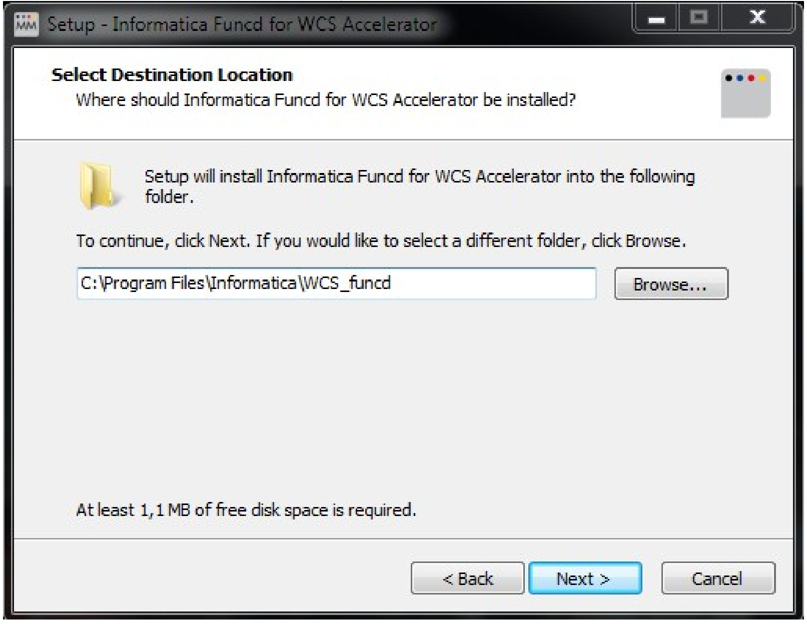
Welcome to the Informatica Funcd for WCS Accelerator Setup Wizard

This will install Informatica Funcd for WCS Accelerator 7.1 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

4. Define the installation directory and click Next



Setup - Informatica Funcd for WCS Accelerator

Select Destination Location

Where should Informatica Funcd for WCS Accelerator be installed?

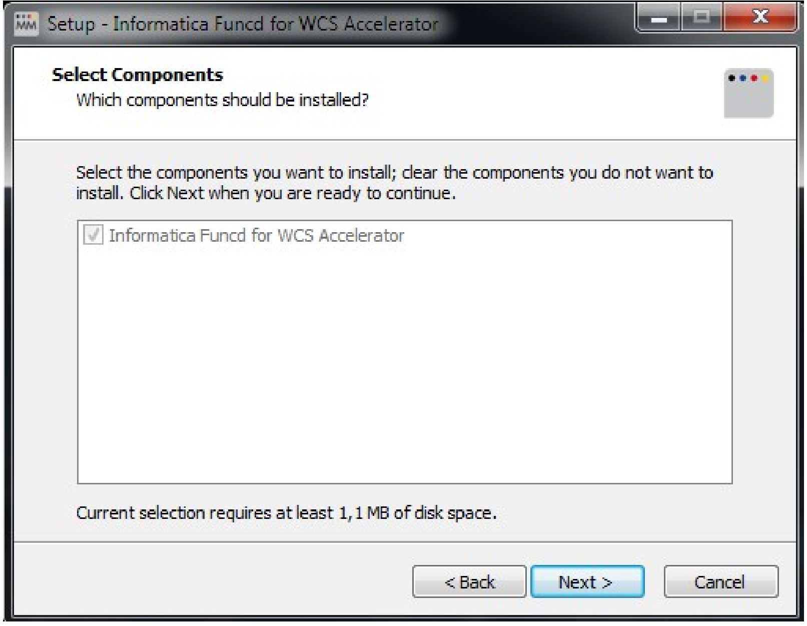
Setup will install Informatica Funcd for WCS Accelerator into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Informatica\WCS_funcd

At least 1,1 MB of free disk space is required.

5. There are no additional components to choose, just click Next



Setup - Informatica Funcd for WCS Accelerator

Select Components

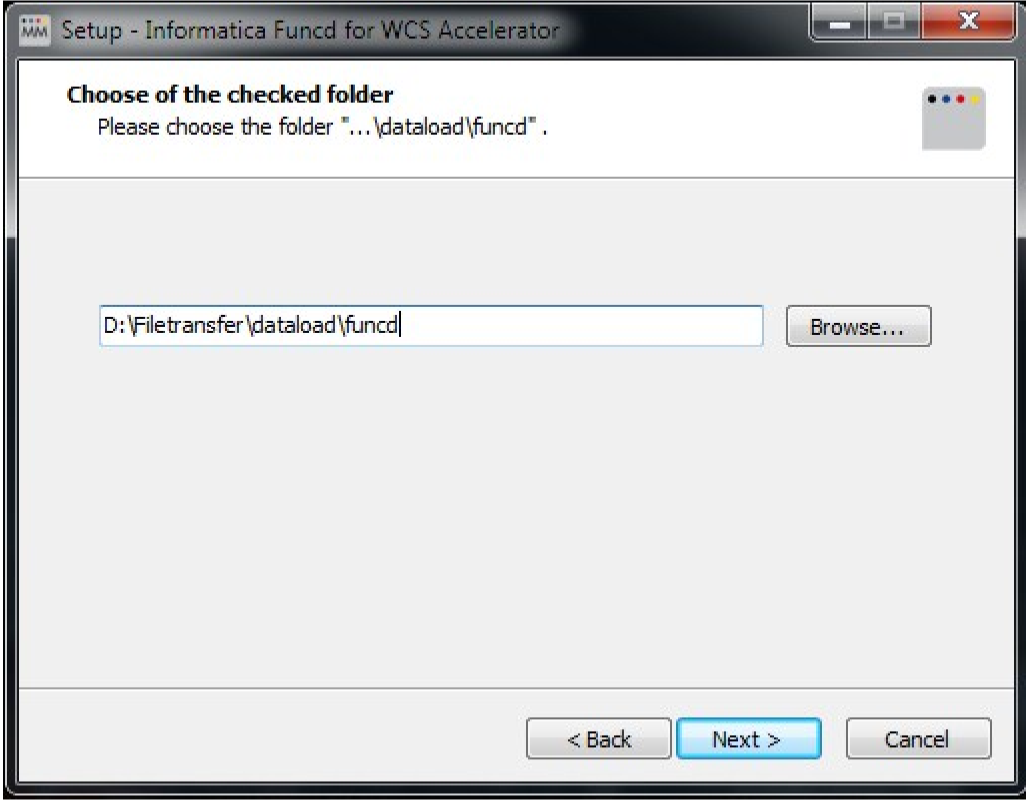
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☑ Informatica Funcd for WCS Accelerator

Current selection requires at least 1,1 MB of disk space.

6. Define the folder which you want to poll. In our example we used **<SHARED_DIRECTORY_PATH>** `\dataload\funcd`. The path must be in the shared folder because the Product 360 export needs a way to start the cmd and bat files locally. Click Next after defining.



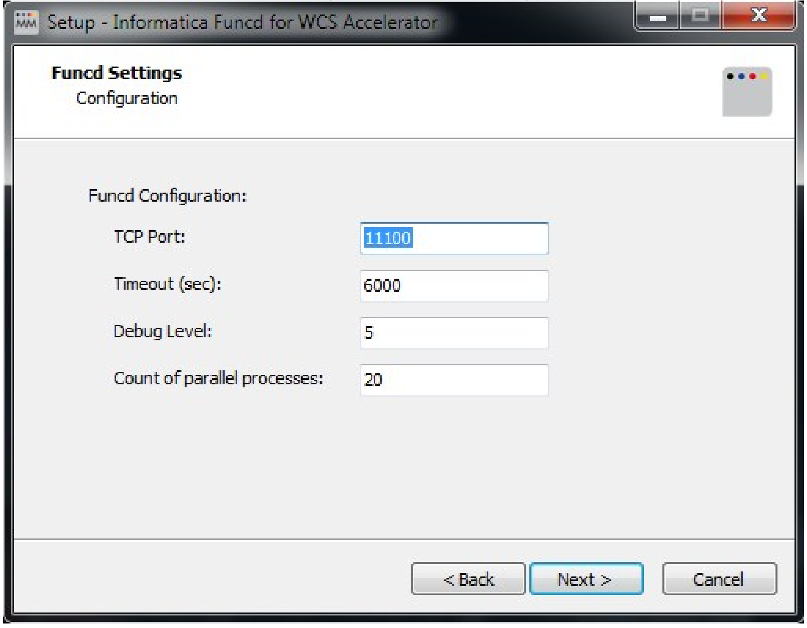
Setup - Informatica Funcd for WCS Accelerator

Choose of the checked folder

Please choose the folder "...\dataload\funcd" .

D:\Filetransfer\dataload\funcd

7. Different possibilities to change the some configuration of the funcd. Click Next



Setup - Informatica Funcd for WCS Accelerator

Funcd Settings

Configuration

Funcd Configuration:

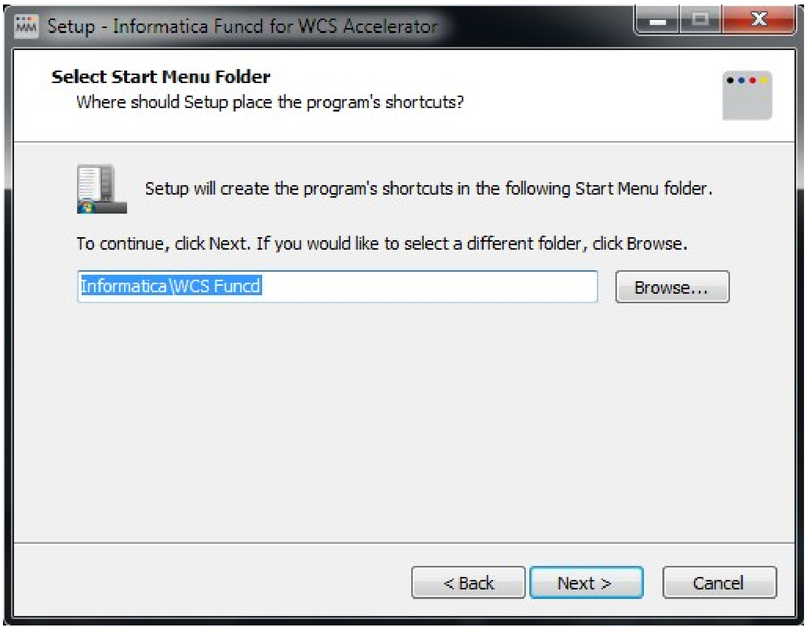
TCP Port: 11100

Timeout (sec): 6000

Debug Level: 5

Count of parallel processes: 20

8. Define the start menu folder and click Next



Setup - Informatica Funcd for WCS Accelerator

Select Start Menu Folder

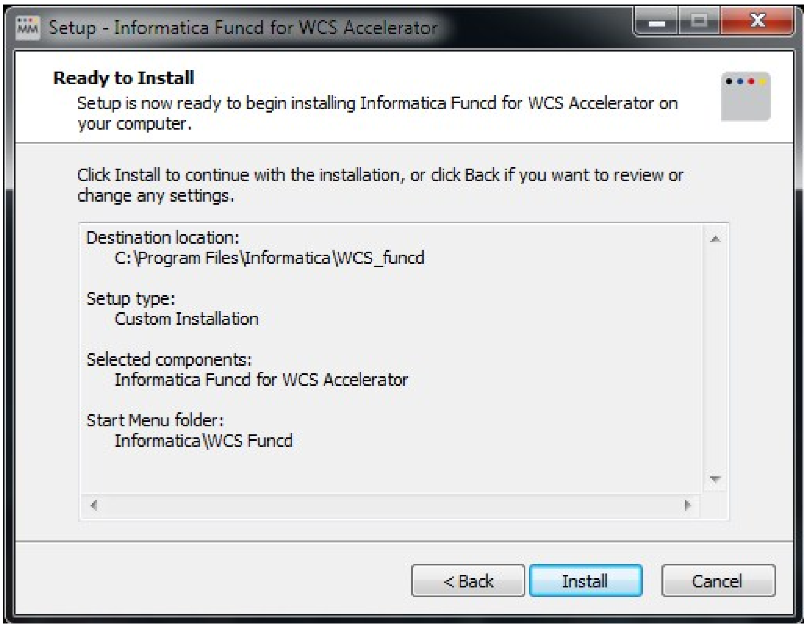
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Informatica\WCS Funcd

9. Review the install overview and click Install



Setup - Informatica Funcd for WCS Accelerator

Ready to Install

Setup is now ready to begin installing Informatica Funcd for WCS Accelerator on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files\Informatica\WCS_funcd

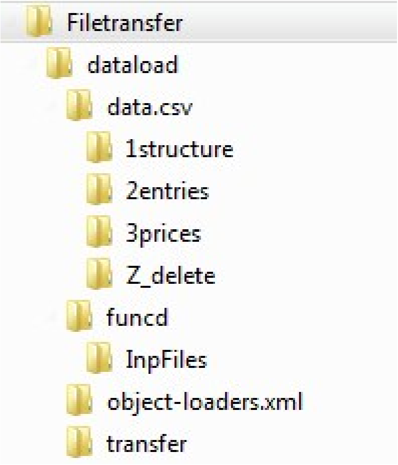
Setup type:
Custom Installation

Selected components:
Informatica Funcd for WCS Accelerator

Start Menu folder:
Informatica\WCS Funcd

10. After a successful installation the tool can be started, There are 3 different ways to do this:
    - Start the tool direct over the Startmenu `/Start/All Programs/Informatica/WCS Funcd/Start WCS Funcd`
      (Attention: This procedure has to be done after every reboot of the server)
    - Start the tool with the scheduled service of the server
    - Create a windows service for the FUNCD. Therefore go to the installed program directory. There are 2 batch files (`MakeService.bat` and `RemoveService.bat`) which creates or removes a service.
      (Attention: Adapt the settings of the service i.e. automatic start)

After all steps above the shared directory should have following structure



- Filetransfer
  - dataload
    - data.csv
      - 1structure
      - 2entries
      - 3prices
      - Z_delete
    - funcd
      - InpFiles
    - object-loaders.xml
    - transfer

2.2.4 Product 360 environment configuration

2.2.4.1 Product 360 server

Unpack the archive **Product 360_<*Version*>_<Revision>_server_websphere_i18n.delta.zip** into the Product 360 server folder.

2.2.4.2 Product 360 client

1. Unpack the archive **Product 360_<*Version*>_<Revision>_client_websphere_i18n.delta.zip** into the Product 360 client folder.
2. Start the client and save the export format templates from the unpacked zip `exporttemplates` folder into the database.
3. The templates variables needs to get adjusted by your own system specific values.
4. The post export steps needs to get adjusted to the **<SHARED_DIRECTORY_PATH>**

3 Attribute Dictionary

3.1 What is an attribute dictionary?

An IBM WebSphere Commerce (WCS) attribute dictionary is a set of attributes and attribute values. The attribute dictionary can contain attributes with predefined values and attributes with assigned values. Predefined value attributes get a set of values that define the list of allowed values for the respective attribute. Assigned value attributes don't define such an allowed values list, assigned values are specified individually for each catalog entry.

Attributes and attribute values are independent from any catalog entries. You can assign attributes from the attribute dictionary to catalog entries (items and products) for use as defining attributes or descriptive attributes.

3.2 How does the attribute dictionary model match the Product 360 attribute model?

3.2.1 The problem

In IBM WebSphere, all attributes of the attribute dictionary have a unique identifier.

In Product 360, we have two levels of (structure/ structure group) attributes:

- the pool (each pool feature has a unique identifier)
- structure group level (one pool feature can be used at several structure groups with different data types, preset values, ...)

--> We need to combine data of the two levels to create as few as possible unique attributes, the algorithm must provide reproducible results.

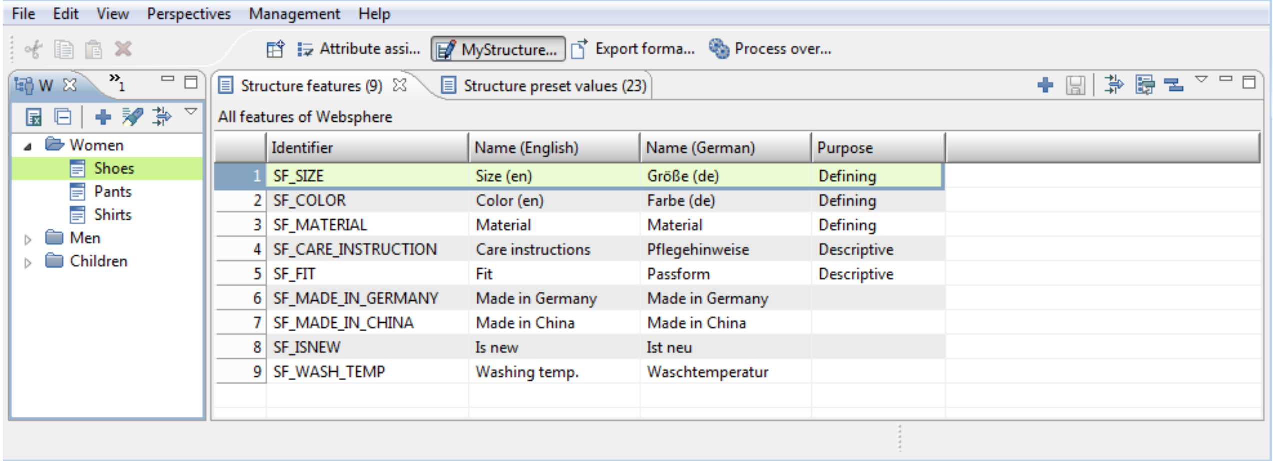
3.2.2 The solution: One pool feature - many structure group attributes - some attributes for the attribute dictionary

We use the following algorithm to collect data for the attribute dictionary:

- collect all structure group attributes
- group them by identifier (it's the identifier of the pool feature) and data type
- cumulate the preset values of each attribute group
- each group will become an attribute of the dictionary, the identifier is <*structure attribute identifier*>_<*id of the data type*>, e.g. **size_1** (the "size" feature with `string` data type), **size_4** (the "size" feature with `integer` data type)

3.2.3 Example: Cumulating attributes

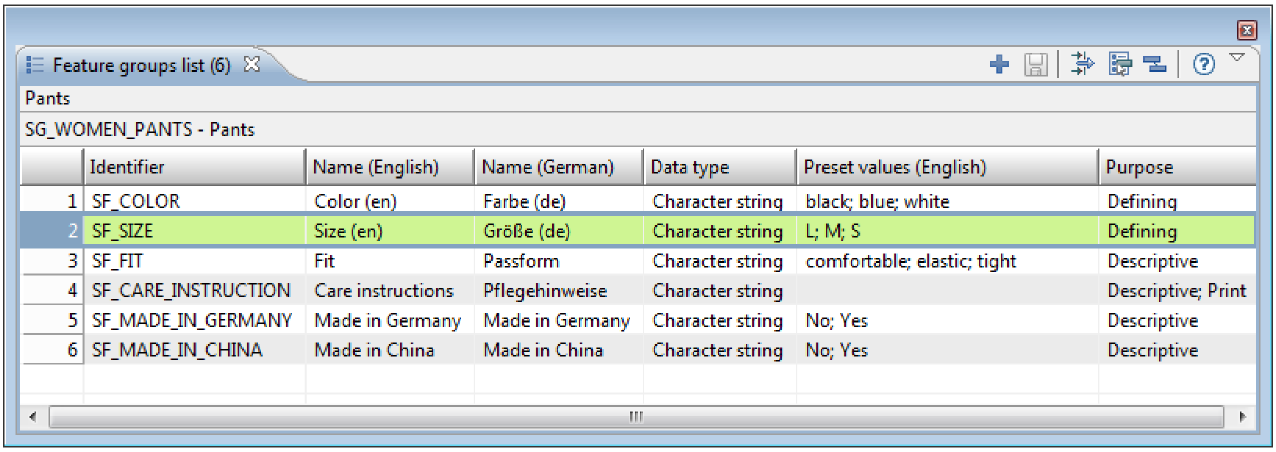
There're some features defined for the structure. One of it is "Size":



All features of Websphere

	Identifier	Name (English)	Name (German)	Purpose
1	SF_SIZE	Size (en)	Größe (de)	Defining
2	SF_COLOR	Color (en)	Farbe (de)	Defining
3	SF_MATERIAL	Material	Material	Defining
4	SF_CARE_INSTRUCTION	Care instructions	Pflegehinweise	Descriptive
5	SF_FIT	Fit	Passform	Descriptive
6	SF_MADE_IN_GERMANY	Made in Germany	Made in Germany	
7	SF_MADE_IN_CHINA	Made in China	Made in China	
8	SF_ISNEW	Is new	Ist neu	
9	SF_WASH_TEMP	Washing temp.	Waschtemperatur	

The "Size" feature is used at the "Pants" structure group as character string data type with preset values:

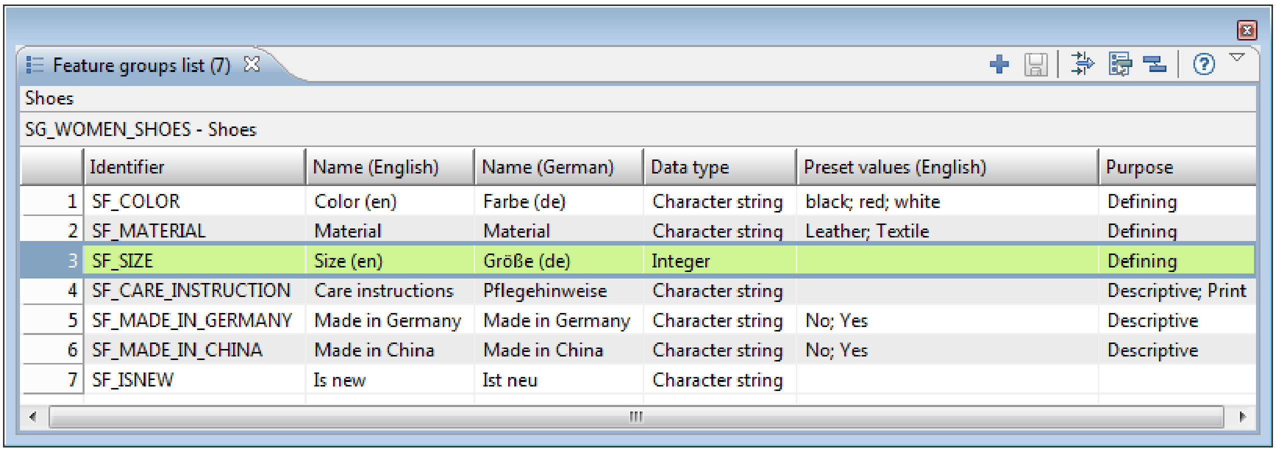


Pants

SG_WOMEN_PANTS - Pants

	Identifier	Name (English)	Name (German)	Data type	Preset values (English)	Purpose
1	SF_COLOR	Color (en)	Farbe (de)	Character string	black; blue; white	Defining
2	SF_SIZE	Size (en)	Größe (de)	Character string	L; M; S	Defining
3	SF_FIT	Fit	Passform	Character string	comfortable; elastic; tight	Descriptive
4	SF_CARE_INSTRUCTION	Care instructions	Pflegehinweise	Character string		Descriptive; Print
5	SF_MADE_IN_GERMANY	Made in Germany	Made in Germany	Character string	No; Yes	Descriptive
6	SF_MADE_IN_CHINA	Made in China	Made in China	Character string	No; Yes	Descriptive

The "Size" feature is used at the "Shoes" structure group as integer data type without preset values:



Shoes

SG_WOMEN_SHOES - Shoes

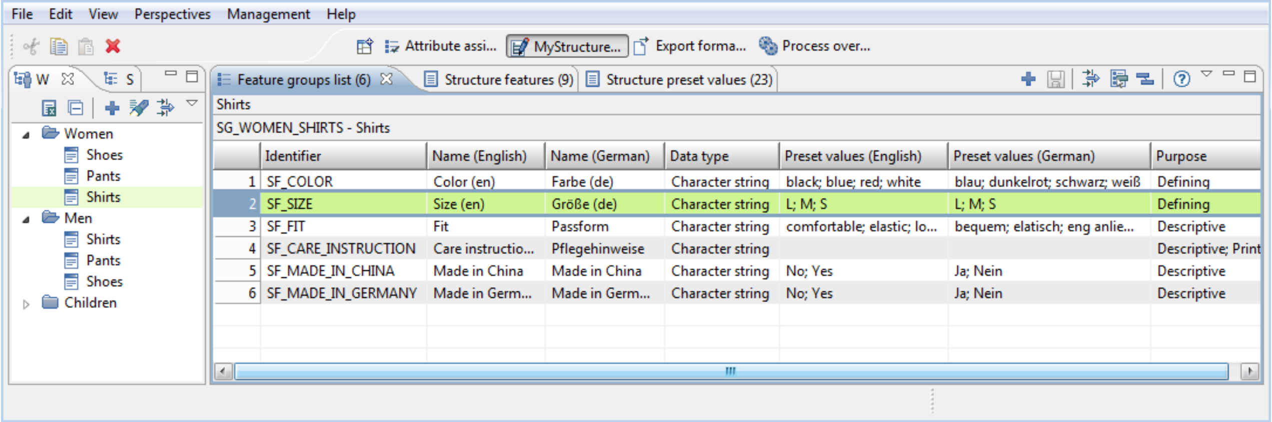
	Identifier	Name (English)	Name (German)	Data type	Preset values (English)	Purpose
1	SF_COLOR	Color (en)	Farbe (de)	Character string	black; red; white	Defining
2	SF_MATERIAL	Material	Material	Character string	Leather; Textile	Defining
3	SF_SIZE	Size (en)	Größe (de)	Integer		Defining
4	SF_CARE_INSTRUCTION	Care instructions	Pflegehinweise	Character string		Descriptive; Print
5	SF_MADE_IN_GERMANY	Made in Germany	Made in Germany	Character string	No; Yes	Descriptive
6	SF_MADE_IN_CHINA	Made in China	Made in China	Character string	No; Yes	Descriptive
7	SF_ISNEW	Is new	Ist neu	Character string		

The algorithm for collecting attribute dictionary data identifies two attributes named "Size":

```
AttributeIdentifier| Type    | AttributeType  | LanguageId| Name
SF_SIZE_1          | STRING  | AllowedValues  | -3        | "Größe"
SF_SIZE_1          | STRING  | AllowedValues  | -1        | "Size"
SF_SIZE_4          | INTEGER| AssignedValues| -3        | "Größe"
SF_SIZE_4          | INTEGER| AssignedValues| -1        | "Size"
```

3.2.4 Example: Collecting preset values of attributes

The "Size" attribute is used at "Women - Shirts" structure group with preset values "L", "M", and "S":

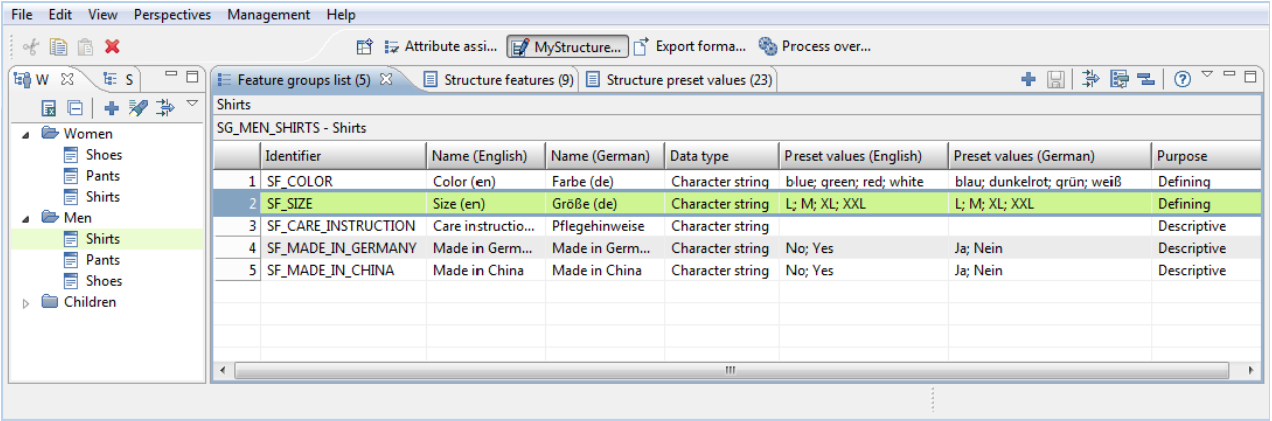


Shirts

SG_WOMEN_SHIRTS - Shirts

	Identifier	Name (English)	Name (German)	Data type	Preset values (English)	Preset values (German)	Purpose
1	SF_COLOR	Color (en)	Farbe (de)	Character string	black; blue; red; white	blau; dunkelrot; schwarz; weiß	Defining
2	SF_SIZE	Size (en)	Größe (de)	Character string	L; M; S	L; M; S	Defining
3	SF_FIT	Fit	Passform	Character string	comfortable; elastic; lo...	bequem; elatisch; eng anlie...	Descriptive
4	SF_CARE_INSTRUCTION	Care instructio...	Pflegehinweise	Character string			Descriptive; Print
5	SF_MADE_IN_CHINA	Made in China	Made in China	Character string	No; Yes	Ja; Nein	Descriptive
6	SF_MADE_IN_GERMANY	Made in Germ...	Made in Germ...	Character string	No; Yes	Ja; Nein	Descriptive

The "Size" attribute is used at "Men - Shirts" structure group with preset values "L", "M", "XL", and "XXL":



Shirts

SG_MEN_SHIRTS - Shirts

	Identifier	Name (English)	Name (German)	Data type	Preset values (English)	Preset values (German)	Purpose
1	SF_COLOR	Color (en)	Farbe (de)	Character string	blue; green; red; white	blau; dunkelrot; grün; weiß	Defining
2	SF_SIZE	Size (en)	Größe (de)	Character string	L; M; XL; XXL	L; M; XL; XXL	Defining
3	SF_CARE_INSTRUCTION	Care instructio...	Pflegehinweise	Character string			Descriptive
4	SF_MADE_IN_GERMANY	Made in Germ...	Made in Germ...	Character string	No; Yes	Ja; Nein	Descriptive
5	SF_MADE_IN_CHINA	Made in China	Made in China	Character string	No; Yes	Ja; Nein	Descriptive

The algorithm for collecting attribute dictionary data cumulates the preset values for the "SIZE_1" attribute:

```
AttributeIdentifier| ValueIdentifier| Value
SF_SIZE_1          | SV_SIZE_S      | "S"
SF_SIZE_1          | SV_SIZE_M      | "M"
SF_SIZE_1          | SV_SIZE_L      | "L"
SF_SIZE_1          | SV_SIZE_XL     | "XL"
SF_SIZE_1          | SV_SIZE_XXL    | "XXL"
```

3.2.5 Limitation

3.2.5.1 Computing the list of dictionary attributes

We only use the structure attribute identifier and the data type of the structure group attributes to compute the list of attributes for the attribute dictionary. We ignore other data that might be necessary in other scenarios, like units.

3.2.5.2 Multi value support

The multi value support for describing catalog entry attributes is disabled to avoid attribute value cumulation.

disabled multi value support in wc-loader-catalog-entry-AD-attribute-relationship.xml

```
<_config:property name="supportMultipleValuesForADAttributes" value="false" />
```

3.2.5.3 Product 360 field - WCS field mapping

Some WCS fields are filled with constant values because there're no matching standard fields in Product 360. For custom solutions it is possible to use appropriate repository reserve fields.

3.2.5.4 Deleted allowed values, deleted attributes

We don't support the export of deleted attributes or attribute allowed values.

3.3 Attributes

The corresponding WCS business object is *AttributeDictionaryAttributeAndAllowedValues*.

Detailed information on the fields can be found in the IBM Knowledge Center.

3.3.1 Types of dictionary attributes

3.3.1.1 Allowed value attributes - assigned value attributes

Dictionary attributes can have a set of allowed values. Assigned value attributes don't define such an allowed values list, assigned values are specified individually for each catalog entry.

The type of an attribute cannot be changed.

3.3.1.2 Defining attributes - describing attributes

The usage of an attribute is specified in the relationship between catalog entries and dictionary attributes.

3.3.2 Export enhancements

The new structure export data provider sub-data type "***Attribute dictionary attributes***" (identifier "ADAttributes") exports data needed for dictionary attributes. It combines several repository entities to collect corresponding data. In addition, it provides two new fields that contain data to fill the "Identifier" and "AttributeType" columns of the "ADAttributes.csv" file. It provides the logical key "Language" of the StructureGroupAttributeLang repository sub-entity as data type filter as well as the attribute "Purpose".

Involved repository entities:

- StructureGroupAttribute
- StructureGroupAttributeLang
- (hidden: StructureGroupAttributePresetValue)
- StructureFeature: transition StructureGroupAttribute.StructureAttribute can be used

Additional provided fields:

- Has preset values --> determine AttributeType "AllowedValues" or "AssignedValues"
- Dictionary attribute identifier

Data type filter:

- Language

- Purpose
- Hidden: Feature name

3.4 Allowed values

The corresponding WCS business object is *AttributeDictionaryAttributeAllowedValues*.

Detailed information on the fields can be found in IBM Knowledge Center.

3.4.1 Export enhancements

The new sub-data type "***Attribute dictionary attribute values***" (Identifier "ADAttributeAllowedValues") of structure export data providers exports data needed for dictionary attribute allowed values. It combines several repository entities to collect corresponding data. In addition, it provides the new "DictionaryAttributeIdentifier" field to fill the "Identifier" column of the "ADAttributeAllowedValues.csv" file. The logical keys "Feature name" and "Preset values" are hidden whereas "Language" and "Purpose" are available data type filters here.

Involved repository entities:

- StructureGroupAttribute
- StructureGroupAttributePresetValue
- StructureValue
- StructureValueLang

Additional provided field:

- Dictionary attribute identifier

Data type filter:

- Language
- Purpose
- Hidden: feature name
- Hidden: Preset value

3.5 Attribute - catalog entry mapping

The corresponding WCS business object is *CatalogEntryAttributeDictionaryAttributeRelationship*.

Detailed information on the fields can be found in IBM Knowledge Center.

The following types of catalog entry to attribute mappings are exported:

1. Defining attributes for items
2. Describing attributes for products
3. Describing attributes for items

It is not necessary to build a relation between a product and the defining attributes used by the child items of the product. Only the item-to-attribute relation has to be considered for defining item attributes.

3.5.1 Export enhancements

3.5.1.1 "GetWCSAttributeUsage" export function

The "GetWCSAttributeUsage" export function calculates the value to be passed as "Usage" column value of the "CatEntryADAttributeValues.csv" file. Return values of that function are "Defining" and "Descriptive".

That function evaluates the attribute purpose given as parameter. It returns "Defining" in case the passed value equals "Defining" or contains "Defining" as one of multiple passed purposes.In any other case "Descriptive" is returned.

Note: The defining attribute purpose has to be defined in the repository enumeration "Enum.AttributePurpose" as enum param with the name "Websphere.defining", current value is "DEFINING".

3.5.1.2 "Attribute dictionary attribute relations" sub-data type

The "***Attribute dictionary attribute relations***" (Identifiers "DataTypeADAttributeArticleRelations", "DataTypeADAttributeProductRelations") export sub-data types are available for all item and product export data providers. They are based on the "Feature values" ("DataTypeArticleFeatureValues" resp. "DataTypeProductFeatureValues") data types. The new data types provide the "Dictionary attribute identifier" field to can identify the dictionary attribute the item resp. product is mapped to.

Involved repository entities:

- ArticleAttribute/ Product2GAttribute
- ArticleAttributeLang/ Product2GAttributeLang
- ArticleAttributeValue/ Product2GAttributeValue

Additional provided field:

- Dictionary attribute identifier

Data type filter:

- Language
- Purpose
- Hidden: attribute name
- Hidden: value identifier

3.6 New and changed attributes and values

The export of new and changed dictionary attributes and allowed values is supported by the "Changed structure data" export data provider.

The following data maintenance operations cause a delta export to grab attribute data during next run:

- create or change structure group attributes
- create or change structure attributes mapped to structure group attributes
- create or change structure values used as allowed value of a structure group attribute

4 Data Exchange

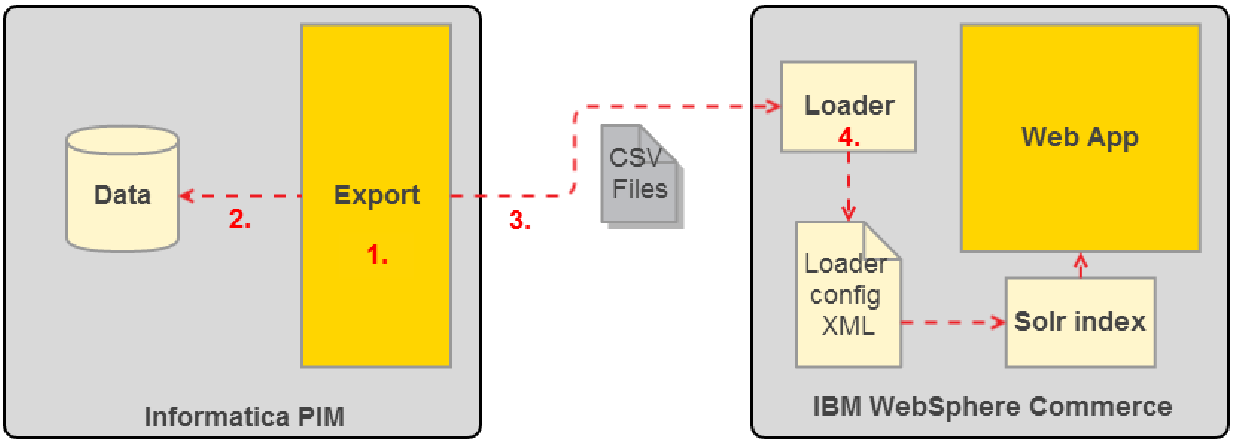
4.1 Scenarios

The WCS Accelerator provides different pre-configured files of export format templates to execute a full export, a delta export or a delta export of a specific product or item. To execute those export format templates, the available variables needs to be configured.

4.1.1 Limitations:

- It is not allowed to change the identifier of product/items/groups because this is the mapping to the WCS data
- It is not allowed to move a catalog group (which is already imported in the WCS) to another parent.
- It is not allowed to change the association type or the referenced entry of an product or item. Please delete the association entry and create a new one instead of changing.

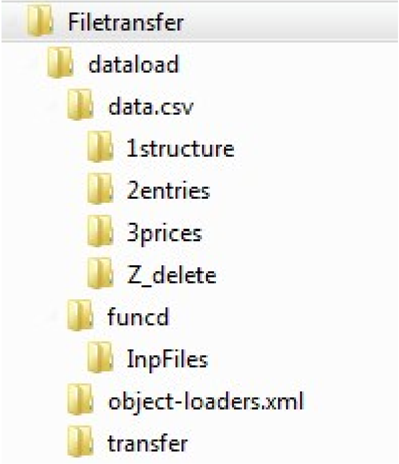
4.2 General Workflow



4.3 WCS Data Loader Environment

4.3.1 Overview

The WCS data loader configurations and working area is provided as directory hierarchy which looks like described below. The root directory `Filetransfer` is a shared folder. This is necessary because the Product 360 export need to transfer the exported files to the WCS environment.



- Filetransfer
  - dataload
    - data.csv
      - 1structure
      - 2entries
      - 3prices
      - Z_delete
    - funcd
      - InpFiles
    - object-loaders.xml
    - transfer

The `dataload` directory contains bat and cmd files, which are starting the image- or data import into the WCS. It also contains WC data load configuration files which define which business object config file of sub-directory `\object-loaders.xml` is mapped to which CSV file of the sub-directory `/data-csv`.

At the beginning of the data load sequence (start_dataLoad_*.cmd) the files are transferred to the directory `\data.csv`. Those CSV files in the sub-directories are used to perform the wcs dataload. (Optional!)The sub-directory \funcd is used to start the dataload from the Product 360 export post export step. The export will move a specific *.inp file in the frequently polled directory. The inp-file will contain the call to a specific cmd file.

The sub-directory `\object-loaders.xml` contains the data loader configuration for the certain CSV data files, where the CSV columns are mapped to the business objects properties for loading.

The sub-directory `\transfer` is the location where the CSV files from the Product 360 export are transferred to initially.

The following icons are used to mark files in the listings below according to their use:

Files marked with this icon	**are needed for...**
★	full load
★	delta load
★	immediate update
★	delete all data

4.3.1.1 Files of the `dataload` directory

★start_dataLoad_full.cmd *(controls the data load sequence for a full import)*
★start_dataLoad_delta.cmd *(controls the data load sequence for a delta import)*
★start_dataLoad_delta_catEntries.cmd *(controls the data load sequence for a product and/or item import)*
★start_dataLoad_deleteAll.cmd *(controls the data load sequence for a full delete)*

🔴🟢🔵cleanupFolders.bat *(deletes all CSV files of the data.csv sub-directories if exist)*
🔴prepareFiles.bat *(copies needed files from transfer directory to the according sub-directory of data.csv)*
🟢🔵prepareDeltaFiles.bat *(copies needed files from transfer directory to the according sub-directory of data.csv)*
🟡prepareDeleteFiles.bat *(copies needed files from transfer directory to the according sub-directory of data.csv)*

🔴🟢load1-structure.bat *(starts loading the catalog groups using wc-dataload-1structure.xml)*
🔴🟢load2-entries.bat *(starts loading the catalog entries using wc-dataload-2entries.xml)*
🔵load2-entries.index.bat *(starts loading the catalog entries using wc-dataload-2entries-searchIndex.xml)*
🔴🟢🔵load3-prices.bat *(starts loading the catalog entries' prices using wc-dataload-3prices.xml)*
🟢🟡loadZ-delete.bat *(starts deleting the catalog groups and entries using wc-dataload-deleteAll.xml)*

🔴🟢🟡buildSOLRindex.bat *(starts rebuild of the SOLr index used by WCS)*
🔵buildSOLRindex_Delta.bat *(starts a delta rebuild of the SOLr index used by WCS)*

🔴🟢🔵🟡wc-dataload-env.xml *(declares the environment variables used by the data loader see The environment configuration file)*
🔴🟢wc-dataload-1structure.xml *(controls the data load sequence for catalog groups see* Data load sequence of structure(see page 24)*)*
🔴🟢wc-dataload-2entries.xml *(controls the data load sequence for catalog entries see* Data load sequence of entries(see page 25)*)*
🔵wc-dataload-2entries-searchIndex.xml *(controls the data load sequence for catalog entries with search index)*
🔴🟢🔵wc-dataload-3prices.xml *(controls the data load sequence for catalog entries' prices see* Data load sequence of prices(see page 26)*)*
🟢🟡wc-dataload-deleteAll.xml *(controls the data load sequence for deleting catalog groups and entries)*

4.3.1.2 Files of the `object-loaders.xml` directory

🔴🟢🟡wc-loader-catalog-group.xml
🔴🟢wc-loader-catalog-group-description.xml

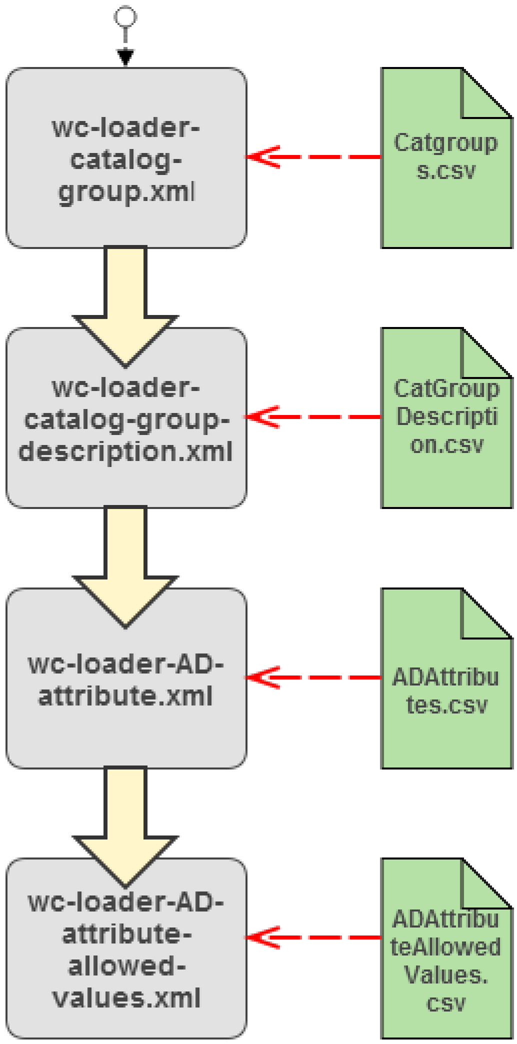
🔴🟢wc-loader-AD-attributes.xml
🔴🟢wc-loader-AD-attribute-allowed-values.xml

🔴🟢🔵🟡wc-loader-catalog-entry.xml
🔵wc-loader-catalog-entry_DELTA.xml
🔴🟢🔵wc-loader-catalog-entry-description.xml
🔴🟢🔵wc-loader-catalog-entry-AD-attribute-relationship.xml
🔴🟢🔵🟡wc-loader-catalog-entry-association.xml
🔴🟢🔵wc-loader-catalogentry-offer-pricelist.xml

4.3.2 WCS object loader mapping

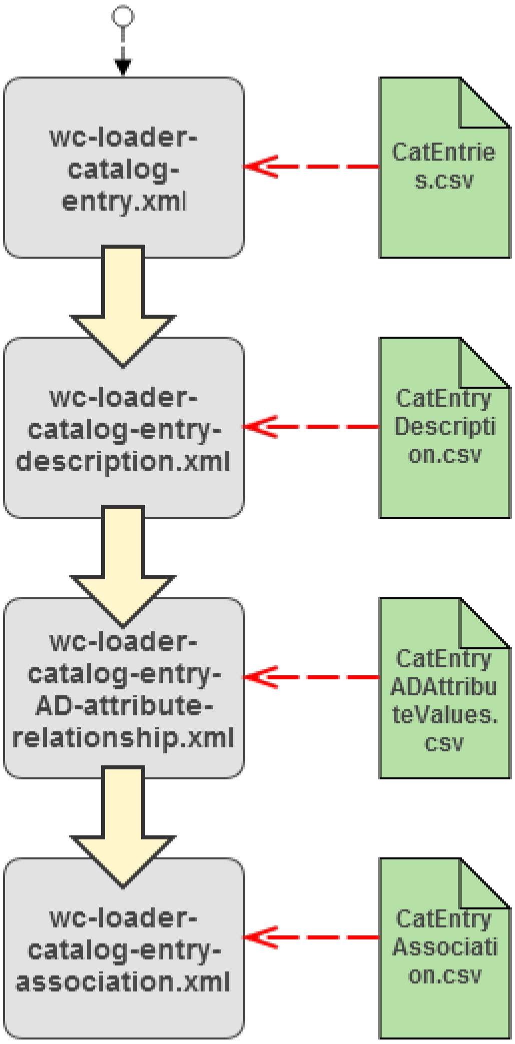
4.3.2.1 Data load sequence of structure

wc-dataload-1structure.xml:


wc-loader-catalog-group.xml
Catgroups.csv
wc-loader-catalog-group-description.xml
CatGroupDescription.csv
wc-loader-AD-attribute.xml
ADAttributes.csv
wc-loader-AD-attribute-allowed-values.xml
ADAttributeAllowedValues.csv

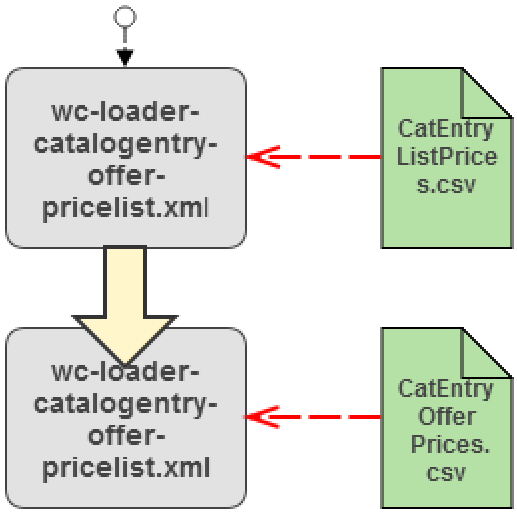
4.3.2.2 Data load sequence of entries

wc-dataload-2entries.xml:


wc-loader-catalog-entry.xml
CatEntries.csv
wc-loader-catalog-entry-description.xml
CatEntryDescription.csv
wc-loader-catalog-entry-AD-attribute-relationship.xml
CatEntryADAttributeValues.csv
wc-loader-catalog-entry-association.xml
CatEntryAssociation.csv

4.3.2.3 Data load sequence of prices

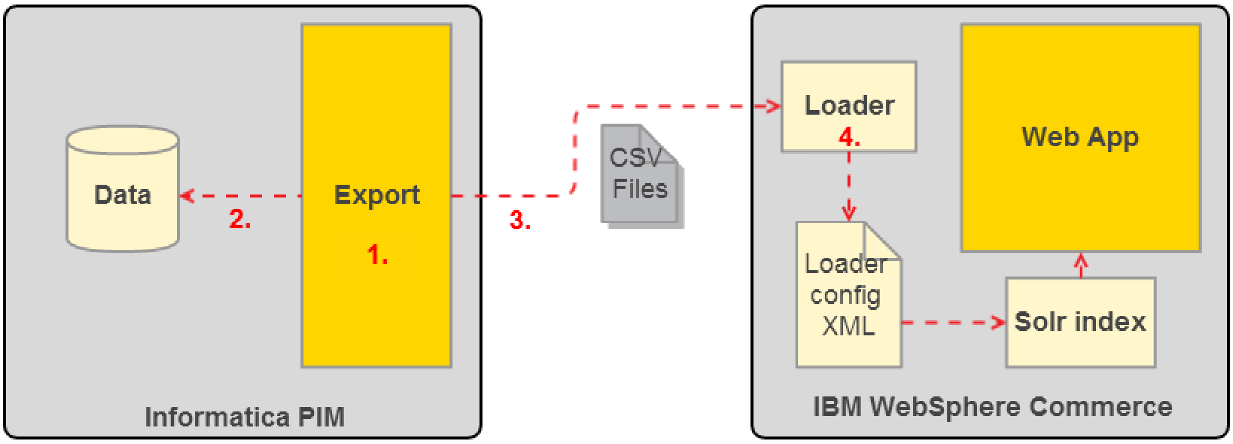
wc-dataload-3prices.xml:


wc-loader-catalogentry-offer-pricelist.xml
CatEntry ListPrices.csv
wc-loader-catalogentry-offer-pricelist.xml
CatEntry Offer Prices.csv

4.4 Workflows

4.4.1 ★Full load

After maintaining the data in a Product 360 system a full export is needed to transfer all necessary data from the Product 360 System to a IBM WebSphere Commerce System. If this transfer is happening the very first time, a full export is needed for this.



1. #### 4.4.1.1 Export template

   The "`WCS - full Export`" export format template is gathering all necessary data and has several post-export step.

2. 4.4.1.2 Exported data

Following data will be exported:
- Collecting data for the Attribute Dictionary.
- Structure group with their language-specific data and file attachments
- Product/Item with their language-specific data, file attachments, referenced products/items, selling price, offer price and attribute values

3. 4.4.1.3 Created Files

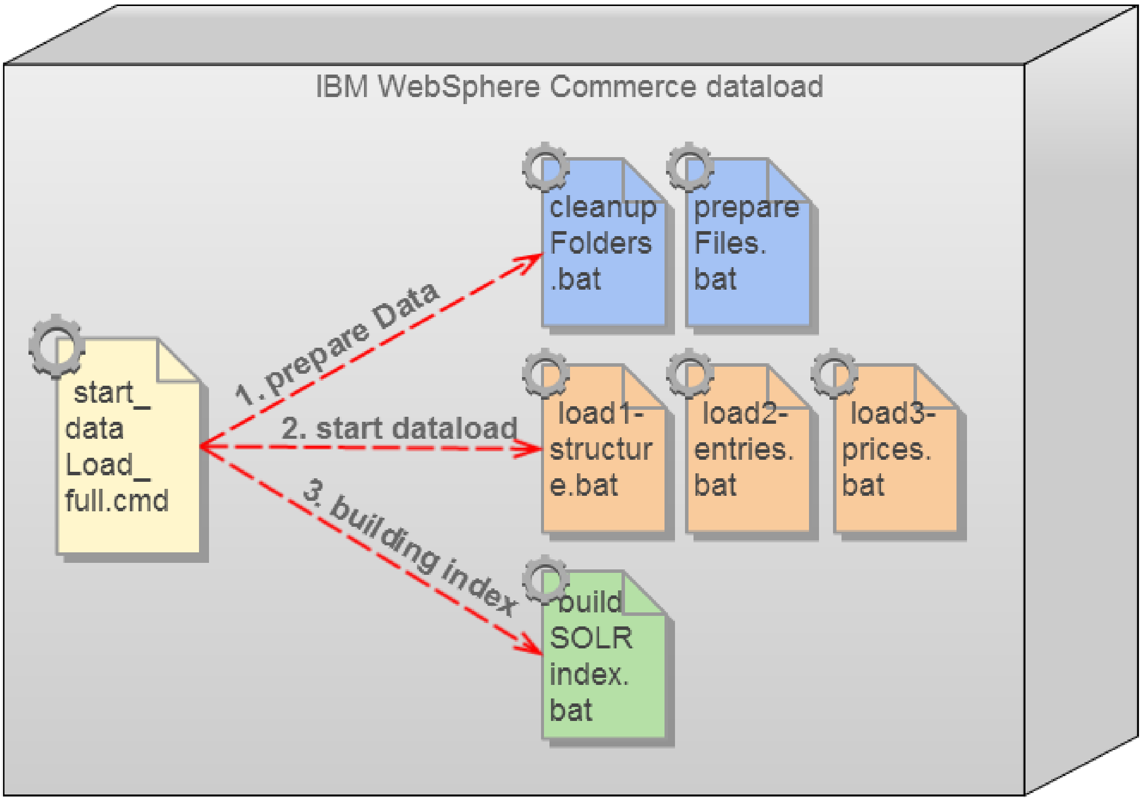
- Multimedia.zip
- ADAttributes.csv
- CatEntries.csv
- CatEntryADAttributeValues.csv
- CatEntryAssociation.csv
- CatEntryDescription.csv
- CatEntryListPrices.csv
- CatEntryOfferPrices.csv
- CatGroupDescription.csv
- CatGroups.csv
- ADAttributeAllowedValues.csv

4. 4.4.1.4 WCS loader

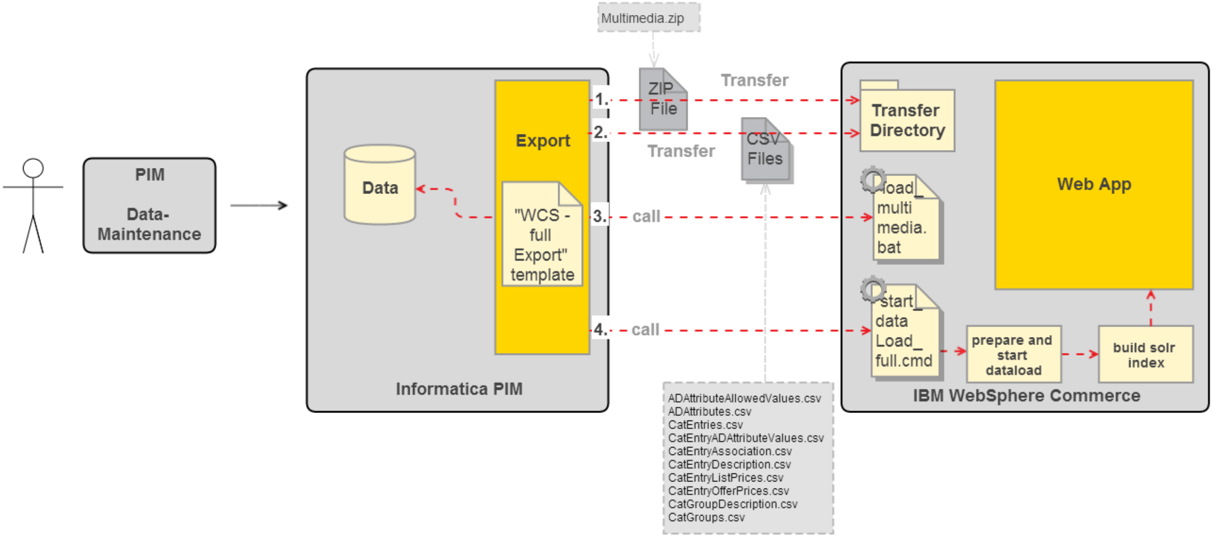
a. Multimedia

`load_multimedia.bat` - will unzip the Multimedia.zip via java command and copy the pictures to the file path. The path is configured in the bat file and probably need to get changed.

b. Data load

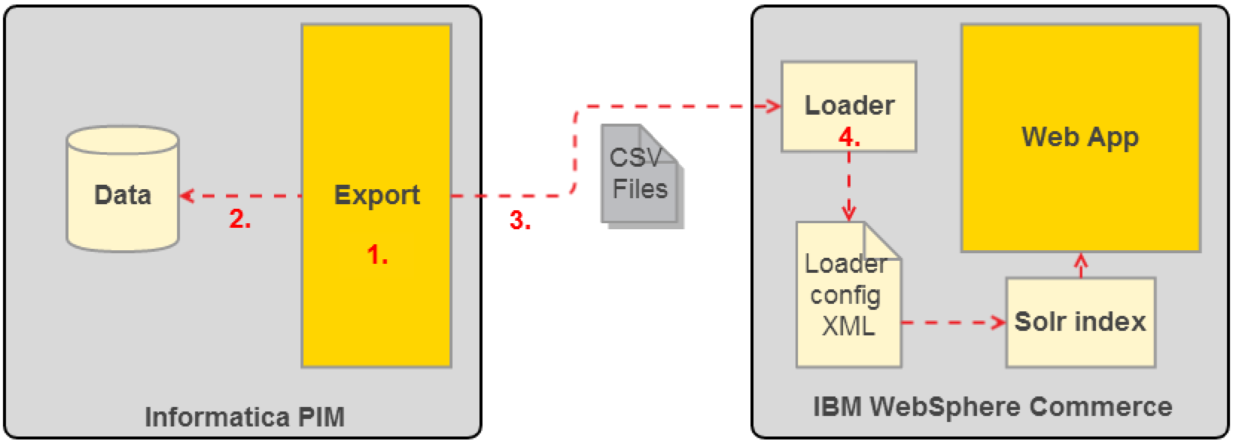


4.4.1.5 Detailed Overview of full export



4.4.2 ★Delta load

After data is already available in the IBM WebSphere Commerce System and maintaining of data in the Product 360 system was continuing a delta export is needed to transfer all new and changed data from the Product 360 System to the mentioned WCS System. This delta update should usually be triggered by a scheduled job.



1. 4.4.2.1 Export template

The "`WCS - Delta Export`" export format template is gathering all necessary deleted, new and changed data and has several post-export step.

2. 4.4.2.2 Exported data

Following data will be exported:
- Products/Items to delete and/or their references, structure group assignment, item to product assignment
- Structure groups to delete
- Collecting data for changed and new Attribute Dictionary.
- Changed and new Structure group with their language-specific data and file attachments
- Changed and new Product/Item with their language-specific data, file attachments, referenced products/ items, selling price, offer price and attribute values

3. 4.4.2.3 Created Files

- Multimedia.zip
- ADAttributes.csv
- CatEntries.csv
- CatEntryADAttributeValues.csv
- CatEntryAssociation.csv
- CatEntryDescription.csv
- CatEntryListPrices.csv
- CatEntryOfferPrices.csv
- CatGroupDescription.csv
- CatGroups.csv
- ADAttributeAllowedValues.csv

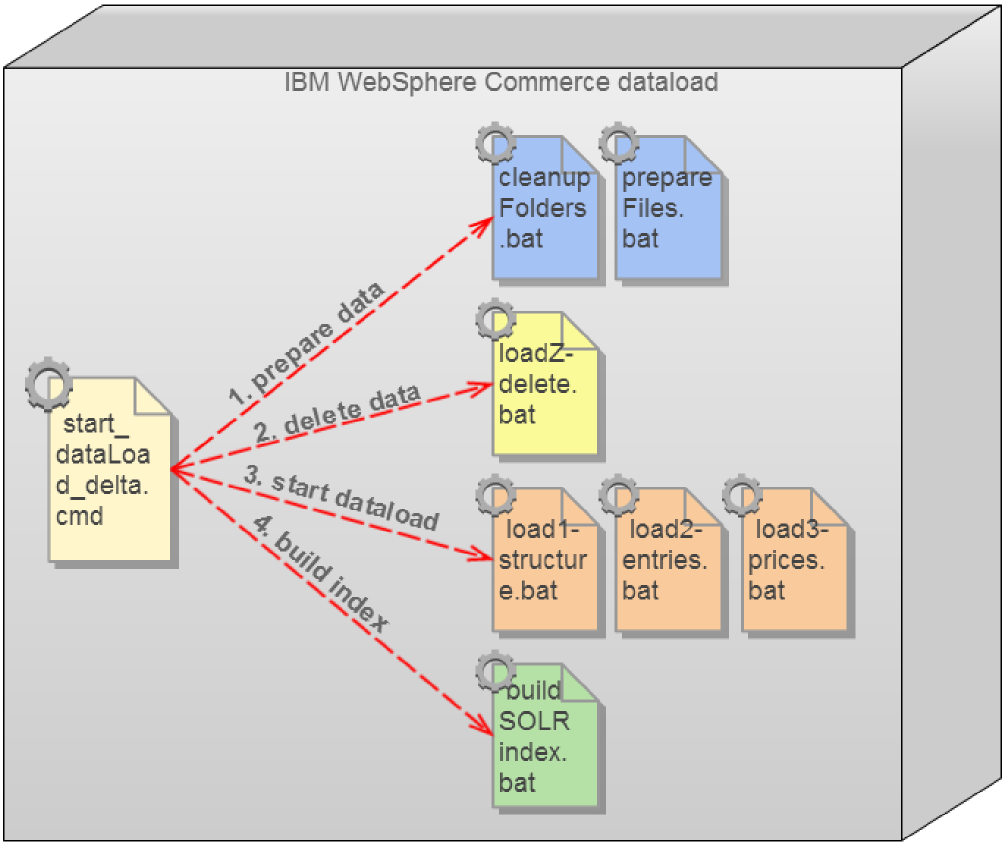
- Z_CatEntriesDelete_Items
- Z_CatEntriesDelete_Products
- Z_CatEntryAssociation
- Z_CatGroupsDelete

4. 4.4.2.4 WCS loader

a. Multimedia

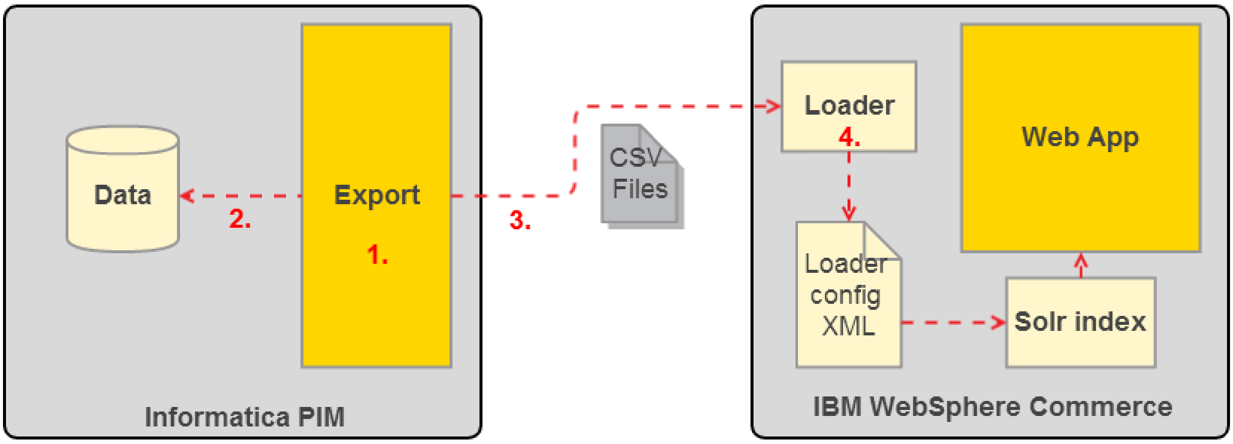
`load_multimedia.bat` - will unzip the Multimedia.zip via java command and copy the pictures to the file path. The path is configured in the bat file and probably need to get changed.

b. Data load



4.4.3 ★Immediate Update

After data is already available in the IBM WebSphere Commerce System and maintaining of data in the Product 360 system was continuing and you want to export some minor changes for an item or a product with its items.



1. ### 4.4.3.1 Export template

   The "`WCS - immediate item export`" export format template is gathering all item data for creating a delta update.
   The "`WCS - immediate product with items export`" export format template is gathering all product and assigned items data for creating a delta update.

2. ### 4.4.3.2 Exported data

   Following data will be exported:
   - Product/Item with their language-specific data, file attachments, referenced products/items, selling price, offer price and attribute values

3. ### 4.4.3.3 Created Files

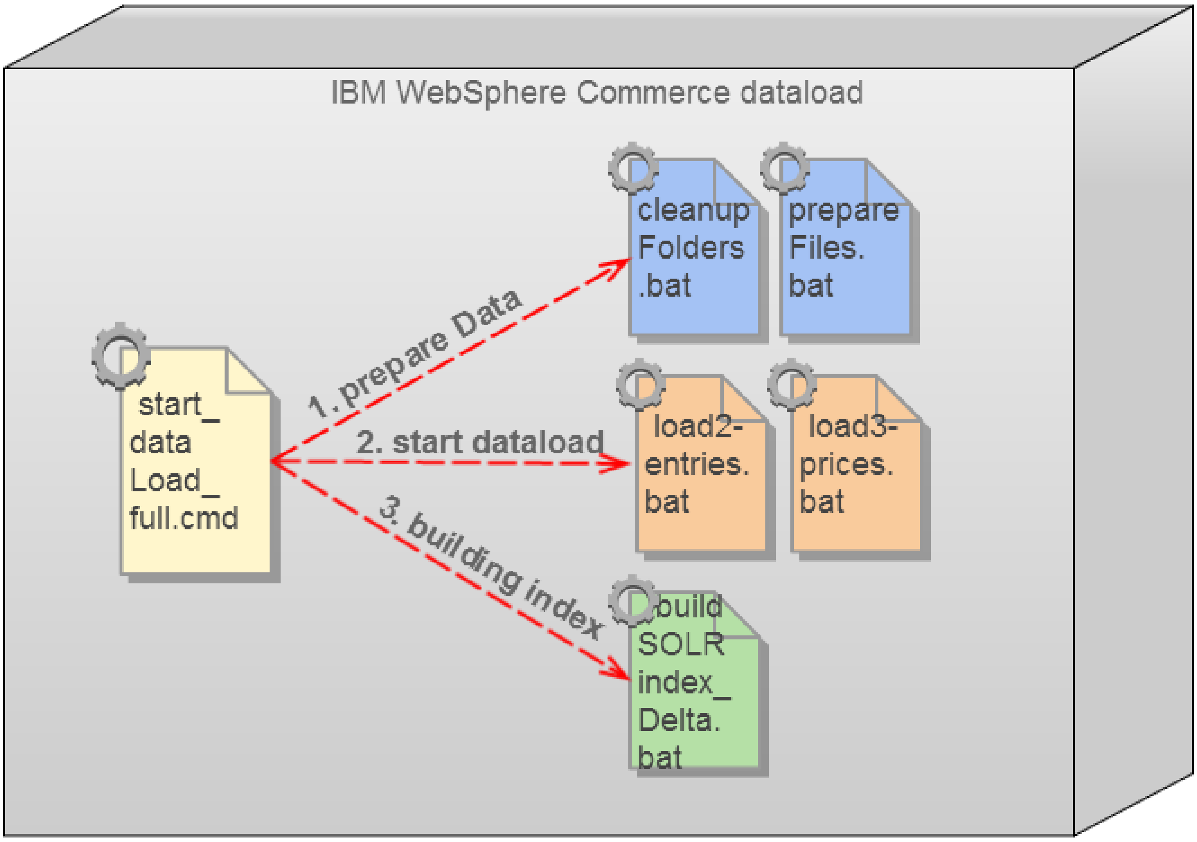
   - Multimedia.zip
   - CatEntries.csv
   - CatEntryADAttributeValues.csv
   - CatEntryAssociation.csv
   - CatEntryDescription.csv
   - CatEntryListPrices.csv
   - CatEntryOfferPrices.csv

4. ### 4.4.3.4 WCS loader

   a. Multimedia

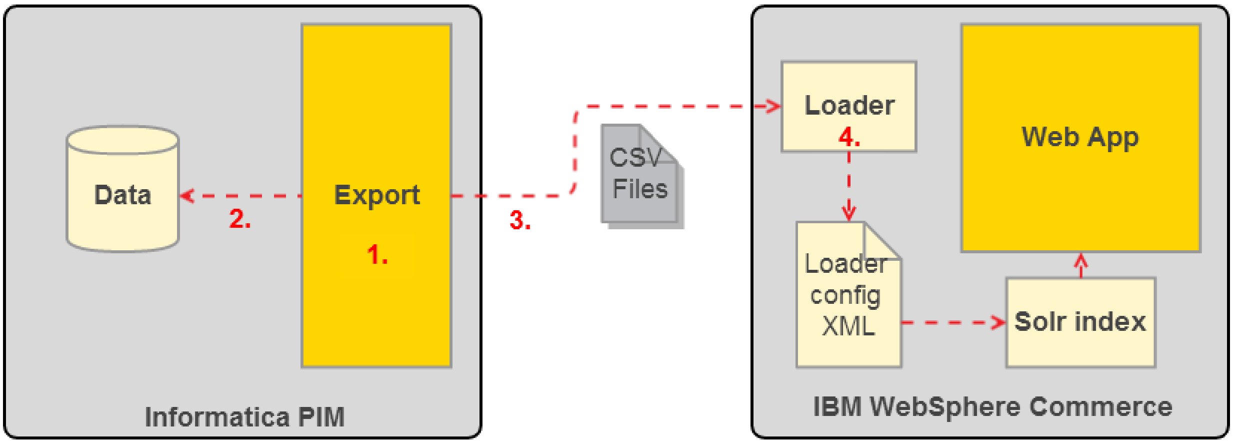
      `load_multimedia.bat` - will unzip the Multimedia.zip via java command and copy the pictures to the file path. The path is configured in the bat file and probably need to get changed.

b. Data load



4.4.4 Delete all data

Specially during testing phase it's important to have the possibility to delete all data on the IBM WebSphere Commerce System. This export format template will export all available data of Product 360 with a deletion flag.



1. 4.4.4.1 Export template

The "`WCS - Delete Items, Products, Structure groups`" export format template is gathering all available data of the Product 360 and marks them as deleted.

2. 4.4.4.2 Exported data

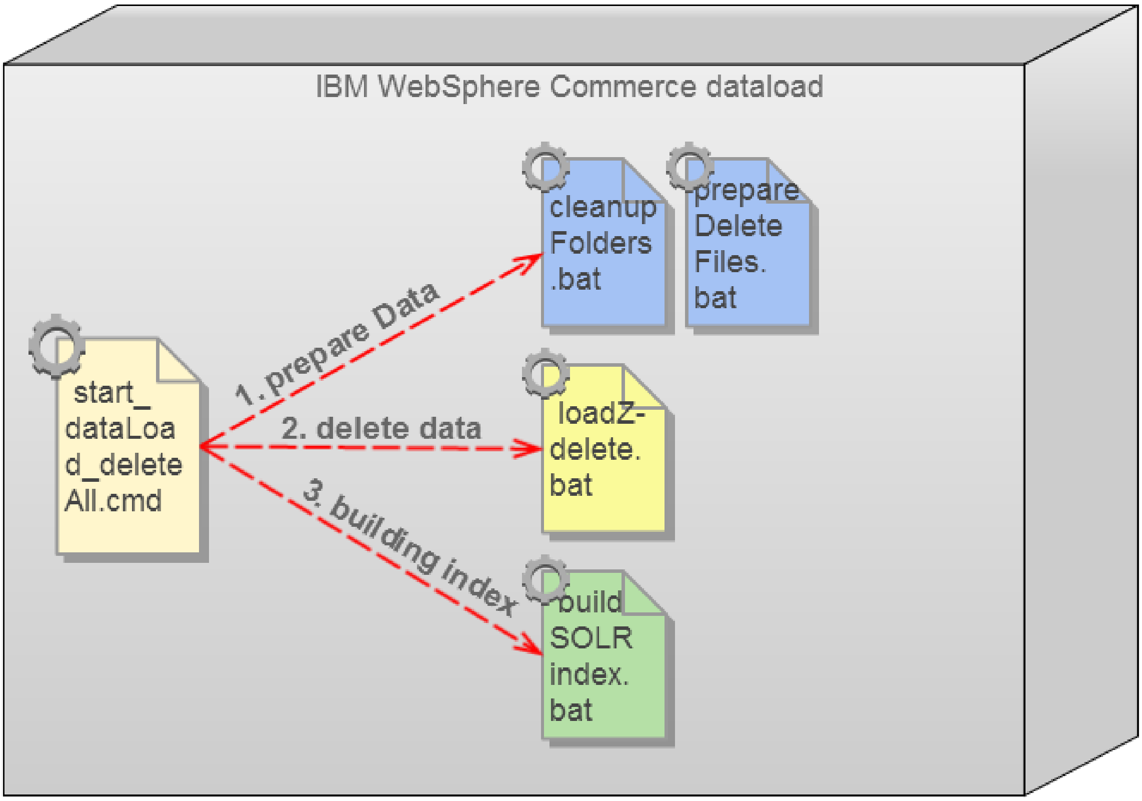
Following data will be exported:
- Structure groups
- Products/Items

3. 4.4.4.3 Created Files

- Z_CatEntriesDelete_Items
- Z_CatEntriesDelete_Products
- Z_CatGroupsDelete

4. 4.4.4.4 WCS loader

   a. Data load



4.5 WCS Data Load Configuration

4.5.1 The data load utility configuration files

The central configuration file for the data load utility consists of two files, the environment configuration file and the data load sequence configuration file.

4.5.1.1 The environment configuration file

See the file `wc-dataload-env.xml` in the root of the data load directory. There are two configuration areas which possibly have to be adapted; the *business context* and the *database connection* parameters.
The business context parameters (element <_config:BusinessContext …>) are:

Attribute	Meaning
storeIdentifier	The store external identifier of the store. The utility resolves the store internal identifier based on the external identifier. Default is: Aurora
catalogIdentifier	The catalog external identifier of the catalog. The utility resolves the catalog internal identifier based on the external identifier. Default is: Aurora

If any of these properties does not match your environment it has to be adapted. Especially the storeIdentifier and/or the catalogIdentifier might be subjects to change.
The database connection settings (element <_config:Database …>) are:

Attribute	Meaning
Type	The database type. Supported values are "derby", "db2" and "oracle". Preset here: "db2"
Name	The name of the database the data is being loaded into. Preset here: DEVFEP7
User	The database user ID. Preset here: db2admin
password	The database password. The database password must be encrypted. You can also leave the password field empty. At the data load runtime, the user will be prompted to enter the password on the command line. Preset here: "".
server	The name of the server the database is on. Preset here: localhost
Port	The port to connect to the database. Preset here: 50000
schema	The name of the database schema. Preset here: db2admin

Refer to the IBM documentation too:
http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.data.doc/code/wc-dataload-env.xsd/WC/xml/config/xsd/wc-dataload-env.xsd_doc/main.html

4.5.1.2 The configuration files for the data load sequence

This configuration contains two main areas of configuration.
First a reference to the environment configuration file (see above) which is provided by the element:
<_config:DataLoadEnvironment configFile=" wc-dataload-env.xml" />
Second the element <_config:LoadOrder ...> where the load items are listed in the sequence the data load should be performed.
A load item contains the mapping between a data file (in our case a CSV file in the directory data) and the corresponding data loader mapping file (an XML file in the directory `object-loaders.xml`). See the sample:

```
<_config:LoadItem name="CatalogGroup" loadSequence="1.0"
businessObjectConfigFile="object-loaders.xml/wc-loader-catalog-group.xml">
<_config:DataSourceLocation location="data.csv/1structure/CatGroups.csv" />
</_config:LoadItem>
```

Additionally the element <_config:LoadOrder ...> supports attributes which allow to control the data load process:

Attribute	Meaning
dataLoadMode	The data load mode. Data load utility modes: • Insert: All data is inserted into the database. The utility generates insert SQL statements. Use this mode for initial data load. • Delete: All data is deleted from the database. The utility generates the delete SQL statements. • Replace (Default): All of the data in the database is replaced. The utility generates the insert, update or delete SQL statements depending on the data. Replace mode replace the existing data with the input data. That is, if some column information is not in the input data, the column value is updated to null or the default value, if any. If defined at the load order level, all load items inherit this value. The value can be overridden at the load item level.
commitCount	Specifies how many lines to process before calling database commit. The default setting is 1. If the commitCount is 0, it means it will not commit until this load item finishes processing all of its input data. If defined at the load order level, all load items inherit this value. The value can be overridden at the load item level.
batchSize	The batch size specifies how many lines of records to process when using JDBC batch. The default value is 1, JDBC batch update is not used. If the batchSize is 0, batches all of processed input data for the entire load item. If defined at the load order level, all load items inherit this value. The value can be overridden at the load item level.

Attribute	Meaning
maxError	This attribute is a setting for the error tolerance during the data load process for a load item. Default is: 1. The utility continues to load data for a load item until the error tolerance level is reached; then the utility exits. Note that for configuration and system errors the utility ends before reaching the tolerance level. If defined at the load order level, all load items inherit this value. The value can be overridden at the load item level.
retry	A flag indicating that if an error occurs during the data load process, the utility retries to load the data. If configuration errors or system errors occur, the utility ends. Default is: false. If defined at the load order level, all load items inherit this value. The value can be overridden at the load item level.

Refer to the IBM documentation too:
http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.data.doc/code/wc-dataload.xsd/WC/xml/config/xsd/wc-dataload.xsd_doc/main.html

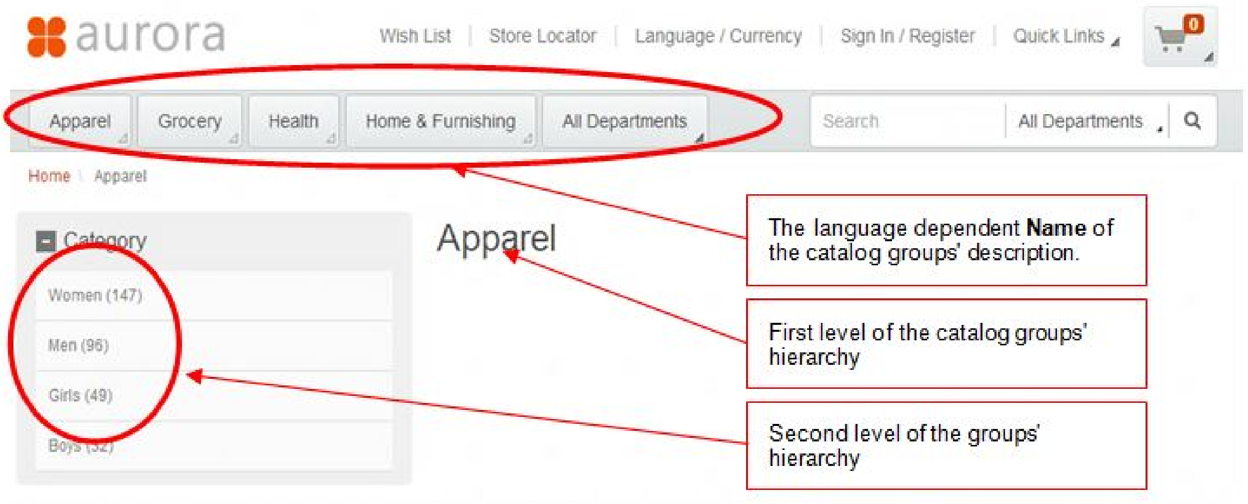
5 Data Maintenance

5.1 Objective

The standard data transfer in the context of the WebSphere Commerce 8 accelerator of Product 360 is optimized for the data entities, functionality and front-end provided by the IBM's sample implementation called Aurora Starter Store.
This documentation provides an overview which data entities are supported by this WebSphere Commerce sample and which data entities and data fields of the standard repository of Product 360 are used to feed the data transfer interface.
Rules and recommendation for the data maintenance are given to fulfil the requisitions of the target system.

5.2 The catalog groups' data maintenance

The IBM WebSphere Commerce platform follows a hierarchical structure system of displaying the different categories in a catalog. It follows the 2-tier hierarchy model.

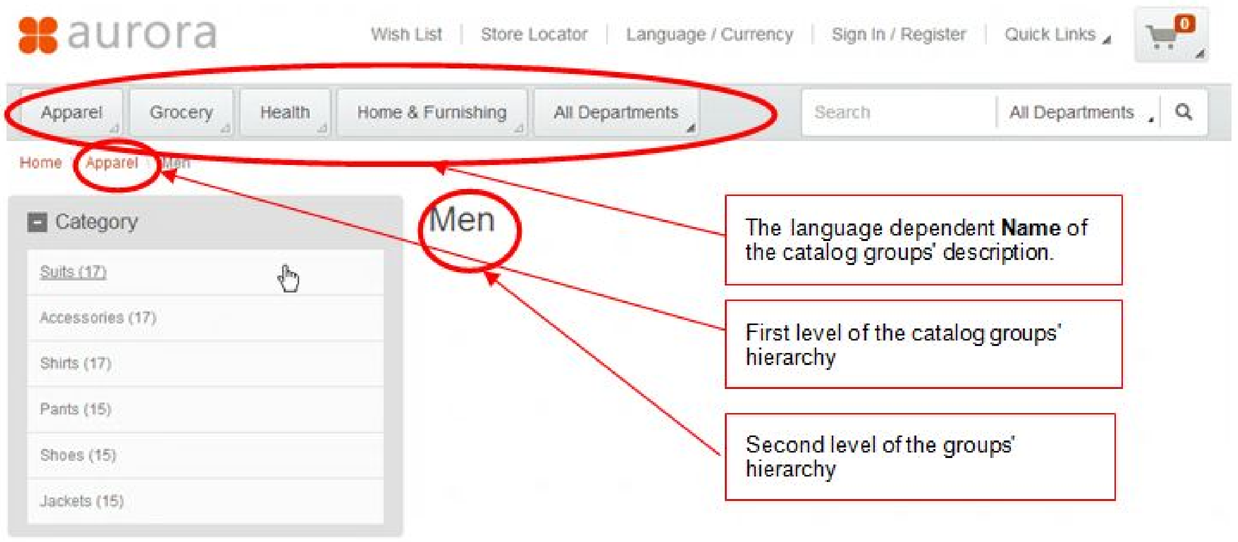


The language dependent **Name** of the catalog groups' description.

First level of the catalog groups' hierarchy

Second level of the groups' hierarchy

Home page of a certain top level group (here Apparel):



The language dependent **Name** of the catalog groups' description.

First level of the catalog groups' hierarchy

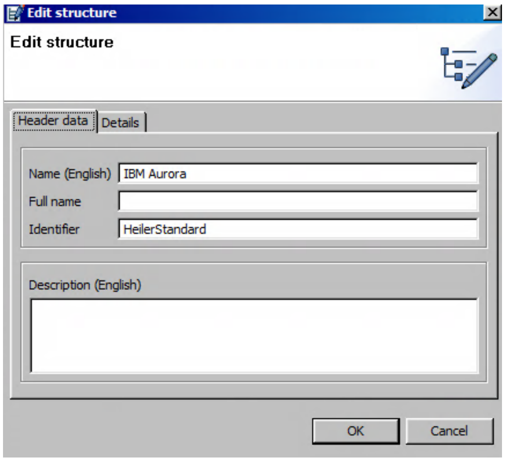
Second level of the groups' hierarchy

Home page of a sub-group (here *Apparel | Men*):

5.2.1 Catalog groups maintenance in Product 360

Basically a structure system has to be provided in Product 360 which represents the Master Catalog of the WebSphere Commerce store.
Creating and maintaining structure systems is a functionality of the perspective *Structures* in the Product

360 client.


Edit structure

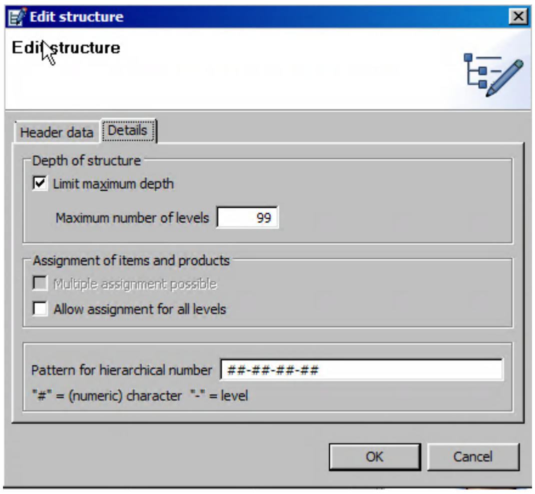
Edit structure

Name (English) IBM Aurora

Full name

Identifier HeilerStandard

Description (English)


Edit structure

Edit structure

Depth of structure

☑ Limit maximum depth

Maximum number of levels 99

Assignment of items and products

☐ Multiple assignment possible

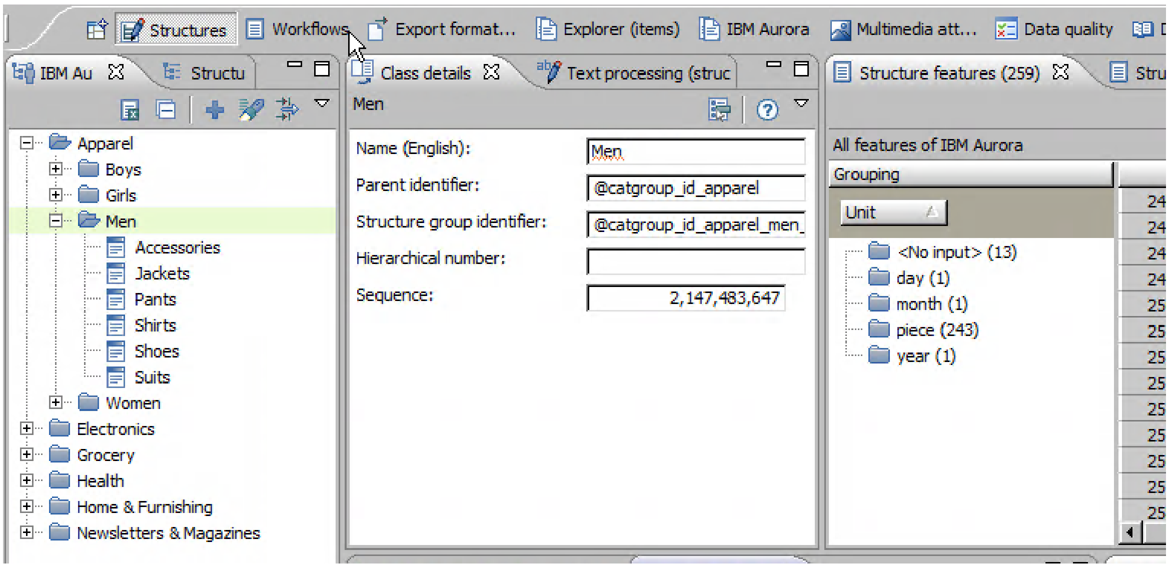
☐ Allow assignment for all levels

Pattern for hierarchical number ##-##-##-##

"#" = (numeric) character "-" = level

A language dependent name and a unique identifier (used in Product 360 only) have to be provided.
The depth of the structure system has to be limited to that level the target system is made for.
The master catalog structure of WebSphere commerce expects a unique product/ item assignment (→ switch off the "*Multiple item assignments possible*") and product/item assignments are only allowed on leaf level (→ switch off the "*Allow item assignment for all levels*").

> The Aurora starter store used as a sample is optimized for a 2-tier hierarchy. The sample data provided also represents a two level hierarchy, but you are not limited to that.



Men

Name (English): Men

Parent identifier: @catgroup_id_apparel

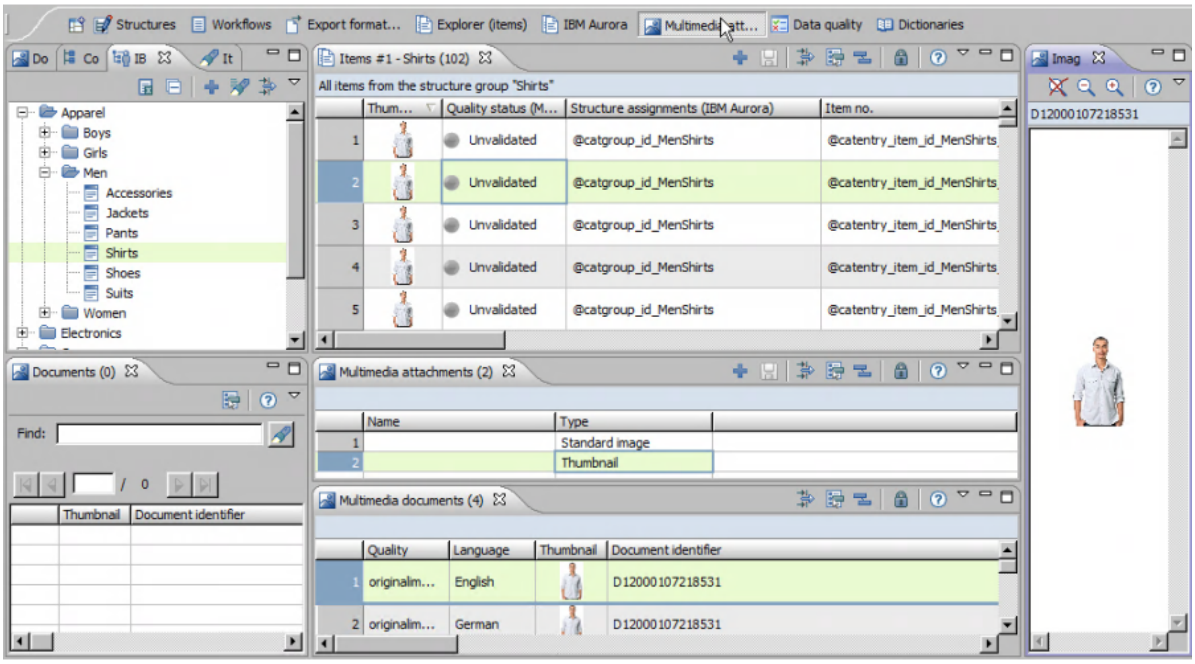
Structure group identifier: @catgroup_id_apparel_men_

Hierarchical number:

Sequence: 2,147,483,647

Hint: The sample screen above shows the group's identifier mapping to the group's name. This is not necessarily needed and only given by the sample data used here.

The structure *group identifier* as well as the *parent identifier* are used in the background to identify the group uniquely, control the parent child relationship, and built the structure's hierarchy. In Product 360 the maintenance of the parent child relationship can easily be done via drag 'n' drop.
Images can be assigned to the structure groups via drag 'n' drop, or specially maintained using the perspective *Multimedia attachments*.



- Apparel
  - Boys
  - Girls
  - Men
    - Accessories
    - Jackets
    - Pants
    - Shirts
    - Shoes
    - Suits
  - Women
- Electronics

Items #1 - Shirts (102)

All items from the structure group "Shirts"

	Thum...	Quality status (M...	Structure assignments (IBM Aurora)	Item no.
1		Unvalidated	@catgroup_id_MenShirts	@catentry_item_id_MenShirts
2		Unvalidated	@catgroup_id_MenShirts	@catentry_item_id_MenShirts
3		Unvalidated	@catgroup_id_MenShirts	@catentry_item_id_MenShirts
4		Unvalidated	@catgroup_id_MenShirts	@catentry_item_id_MenShirts
5		Unvalidated	@catgroup_id_MenShirts	@catentry_item_id_MenShirts

Multimedia attachments (2)

	Name	Type
1		Standard image
2		Thumbnail

Multimedia documents (4)

	Quality	Language	Thumbnail	Document identifier
1	originalim...	English		D12000107218531
2	originalim...	German		D12000107218531

	Thumbnail	Document identifier

Perspective Multimedia attachments used for the maintenance of structure groups images.

Images are generally maintained language dependent in Product 360 and WCS as well. The following table shows the image settings to be used; pre-defined by the standard WebSphere Commerce data export.

WCS Image	Product 360 image type	Product 360 image quality	Description
Full image	Standard image	Internet image	Provide a full image for structure groups using the Product 360 image type *Standard image* and the HPM image quality *Internet image*. The visualization of these images in the Aurora starter store sample is optimized for for images of 588px width and a height between 130px and 160px

Thumbnail	Thumbnail		Provide a thumbnail image for structure groups using the Product 360 image type *Thumbnail* and the Product 360 image quality *Internet image*. The visualization of these images in the Aurora starter store sample is optimized for 192px width and a height of about 94px (for sub-groups, 142px × 118px top level groups).

The data field *Sequence* controls the display order of the top level groups and the sub-ordinate groups below its parent.
In Product 360 the sequence of the top level groups can be maintained simply by drag 'n' drop, or by editing the numeric sequence value (a whole number) in the class details view in relation to the other top level groups.

5.2.2 Data fields of structure groups used for the standard interface

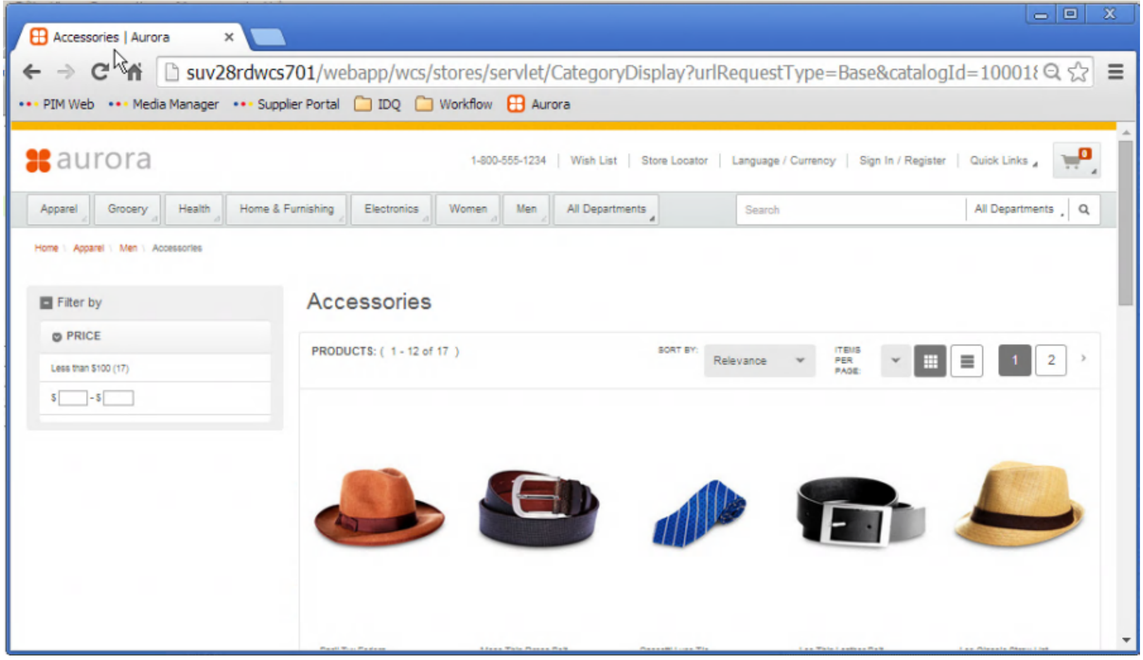
Product 360 data field		**Note**
Structure group identifier (Rep. Ident.: StructureGroup.Identifier)	GroupIdentifier	The unique group identifier of the group. (Mandatory, 254 bytes* maximum)
Parent group identifier (Rep. Ident.: StructureGroup.ParentId)	ParentGroupIdentifier	The unique group identifier of the parent group. (Mandatory, 254 bytes* maximum)
Sequence (Rep. Ident.: StructureGroup.DisplayOrder)	TopGroupSequence Sequence	Controls the display order of the groups below their parent. Optional
Language Dependent Texts		
Name (English) Name (German) Name (…) (Rep. Ident.: StructureGroupLang.Name)	Name	Language dependent name of the structure group. (Mandatory, cut to 254 bytes* maximum)
Language Dependent Images		

Multimedia attachments of the Structure group for each language The data field used is the Document identifier which is concatenated to a relative image URL. (Rep. Ident.: StructureGroupMediaAssetDocument.Identifier)	FullImage Thumbnail	In Product 360 file attachments are identified by several key values. These are the language, the image type and quality. Images have to be maintained for all languages explicitly even they are not language dependent in their presentation. Use the following key values for the WCS accelerator: For Full Image: Quality: Internet image Type:Standard image For Thumbnail: Quality: Internet image Type:Thumbnail The documentidentifier id mandatory, 239 bytes* maximum, if the checks fail the image data set is removed.

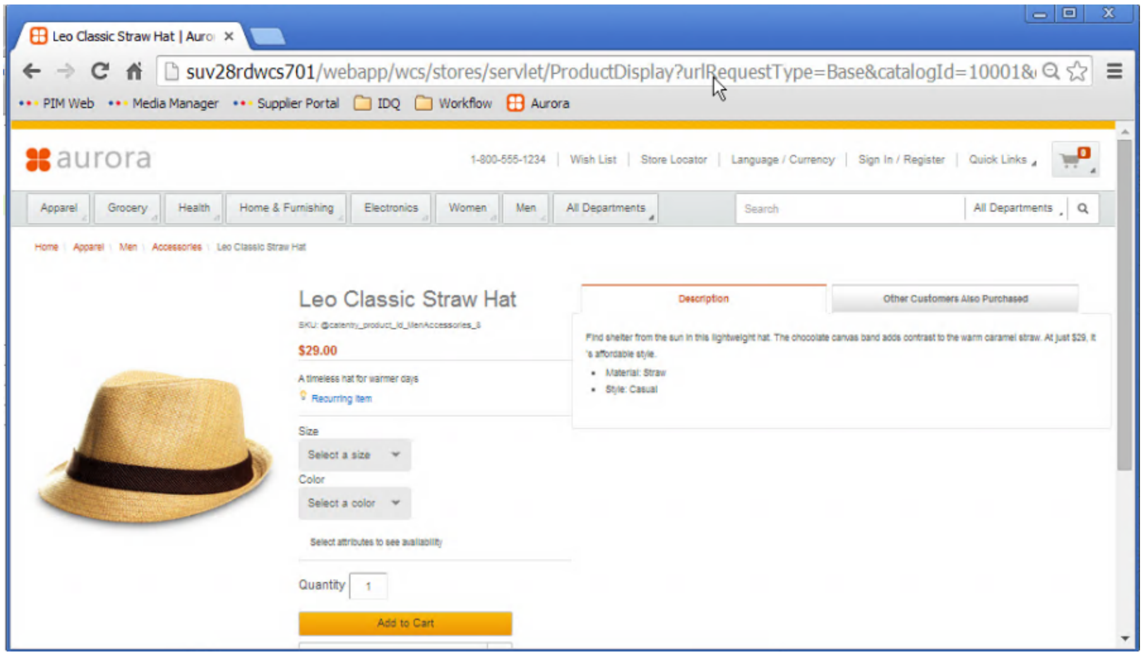
* Most of the WebSphere Commerce text containing data base fields in Oracle are defined as varchar2 (nnn BYTE). Due UTF8 text coming from Product 360 can have 1 to 3 bytes per character this text has to be checked and trimmed by special algorithm to the byte length without damaging the UTF-8 characters.

5.3 The catalog entries' data maintenance

Home page of a sub-group (here Apparel | Accessories) having items:



The detail page of an item:



Leo Classic Straw Hat

SKU: @catentry_product_id_MenAccessories_8

$29.00

A timeless hat for warmer days

Recurring item

Select attributes to see availability

Find shelter from the sun in this lightweight hat. The chocolate canvas band adds contrast to the warm caramel straw. At just $29, it 's affordable style.

- Material: Straw
- Style: Casual

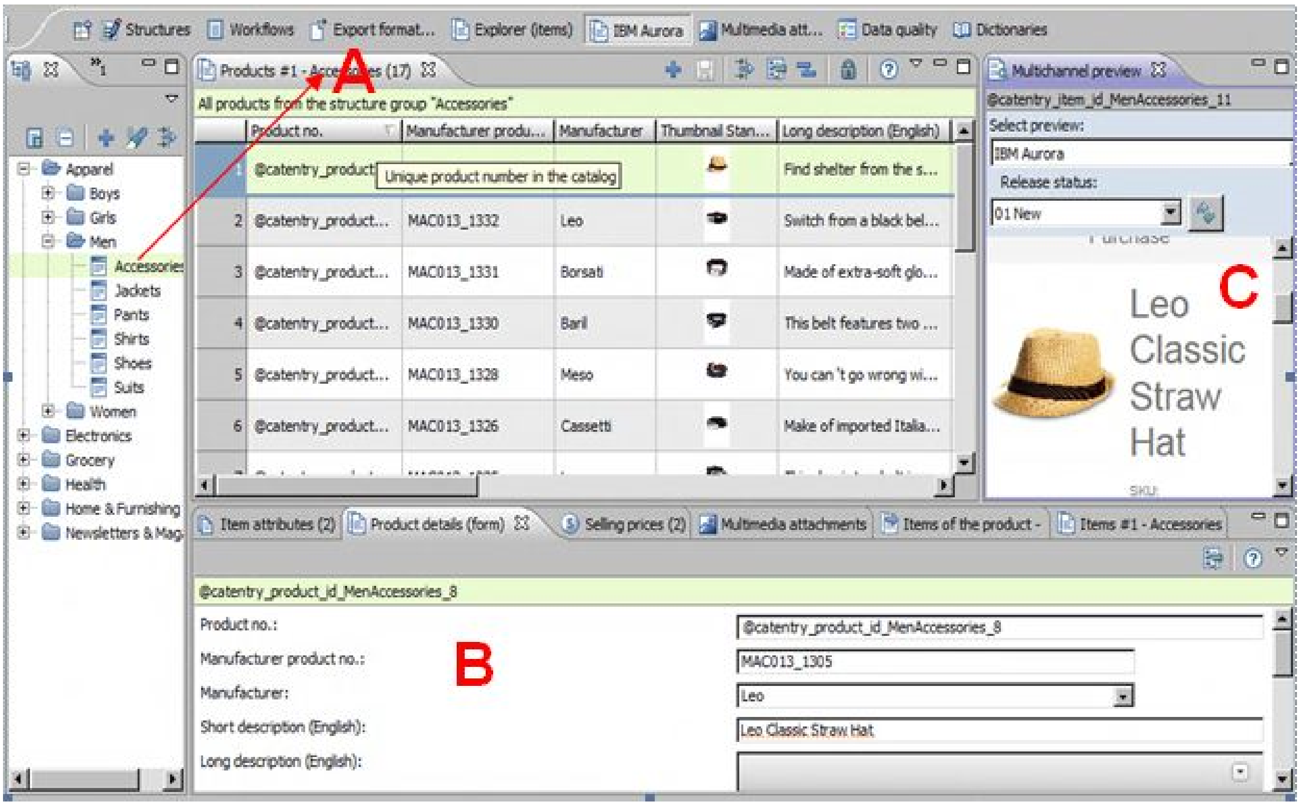
Based on the structure in the commerce site, products and items – have to be maintained with text and image information. This allows having a more common description for the product (valid for all variants) and additionally individual and specific text or image information for the variants.

Products and items maintenance in Product 360

Each product is uniquely assigned to the Product 360 structure system which is used as Master Catalog for the WebSphere Commerce Shop.

The product view *A* in our sample below shows the products of the category *Accessories* selected in the tree on the left.

The detail views *B* visualize the language dependent text information of the product. These are the unique product number, manufacturer number, a short and a long description. The detail view C shows the preview of the product and its items.



All products from the structure group "Accessories"

	Product no.	Manufacturer produ...	Manufacturer	Thumbnail Stan...	Long description (English)
1	@catentry_product				Find shelter from the s...
2	@catentry_product...	MAC013_1332	Leo		Switch from a black bel...
3	@catentry_product...	MAC013_1331	Borsati		Made of extra-soft glo...
4	@catentry_product...	MAC013_1330	Baril		This belt features two ...
5	@catentry_product...	MAC013_1328	Meso		You can 't go wrong wi...
6	@catentry_product...	MAC013_1326	Cassetti		Make of imported Italia...

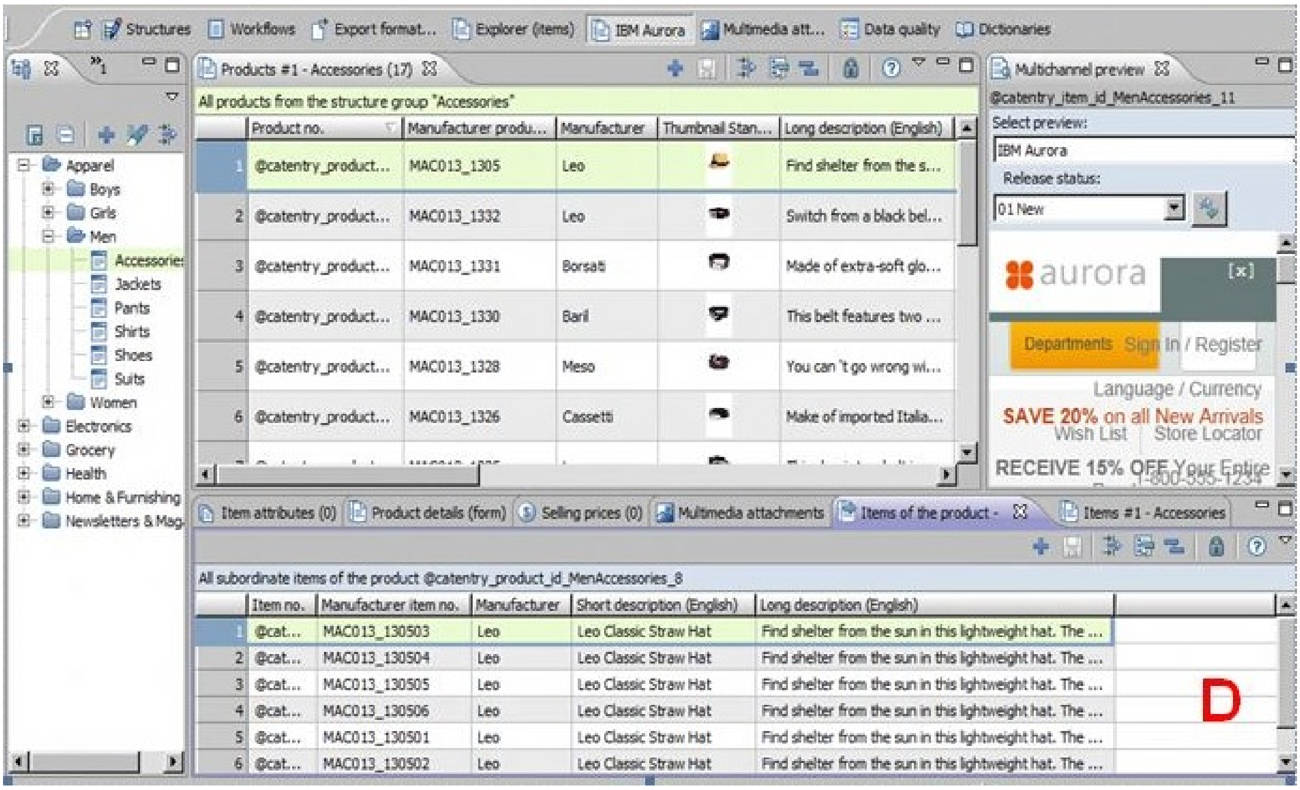
@catentry_item_id_MenAccessories_11

Select preview: IBM Aurora

Release status: 01 New

@catentry_product_id_MenAccessories_8

Product no.:	@catentry_product_id_MenAccessories_8
Manufacturer product no.:	MAC013_1305
Manufacturer:	Leo
Short description (English):	Leo Classic Straw Hat
Long description (English):	



All products from the structure group "Accessories"

	Product no.	Manufacturer produ...	Manufacturer	Thumbnail Stan...	Long description (English)
1	@catentry_product...	MAC013_1305	Leo		Find shelter from the s...
2	@catentry_product...	MAC013_1332	Leo		Switch from a black bel...
3	@catentry_product...	MAC013_1331	Borsati		Made of extra-soft glo...
4	@catentry_product...	MAC013_1330	Baril		This belt features two ...
5	@catentry_product...	MAC013_1328	Meso		You can 't go wrong wi...
6	@catentry_product...	MAC013_1326	Cassetti		Make of imported Italia...

@catentry_item_id_MenAccessories_11

Select preview: IBM Aurora

Release status: 01 New

All subordinate items of the product @catentry_product_id_MenAccessories_8

	Item no.	Manufacturer item no.	Manufacturer	Short description (English)	Long description (English)
1	@cat...	MAC013_130503	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
2	@cat...	MAC013_130504	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
3	@cat...	MAC013_130505	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
4	@cat...	MAC013_130506	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
5	@cat...	MAC013_130501	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
6	@cat...	MAC013_130502	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...

The view *D* shows the associated items for the selected product. As one can see that the attributes defined at the product level are inherited by the subordinate items. The purpose of these product attributes has to be Descriptive. In the Aurora Starter Store on product level the descriptive attributes are shown below the long description on an informational level.
The purpose of these item attributes has to be Defining. The defining attributes are used to select the concrete variant (size, color, etc.) of the product to be ordered. Each of the items pertaining to the different variant is displayed in view D.
Any edits to the attribute values in Product 360 will be reflected in the Aurora Store.

5.3.1 Catalog Entries Description

5.3.1.1 Language dependent texts

In the WebSphere Commerce three main texts of the catalog entry are displayed in different situations on different pages – name, short description and long description.
Three text fields of the standard repository of Product 360 are used and mapped to the WCS text information by the standard data transfer – the *Short description*, the *Other remarks*, and the *Long description*.

See the mapping in the following table:

HPM data field	**WCS data field**
Product/Item Short description (Rep. Ident.: ArticleLang.DescriptionShort)	Catalog Entry Name
Product/Item Other remarks (Rep. Ident.: ArticleLang.Remarks)	Catalog Entry Short Description
Product/Item Long description (Rep. Ident.: ArticleLang.DescriptionLong)	Catalog Entry Long Description

5.3.1.2 Images

In the WebSphere Commerce two main images are used – a thumbnail and a full image. There are some more places, e. g. the shopping cart and the wish list where product or item images are displayed too.
Product 360 allows assigning multiple images (or other file attachments) to an item or a product. These are differentiated by quality and type; and are all language dependent.
For the thumbnail image and the full image used by the WebSphere Commerce Shop a corresponding type/ quality combination has to be defined front up. This has to be considered for data maintenance.
See the following table for the pre-defined settings for the thumbnail and full image used by the standard data transfer for WebSphere Commerce:

WCS Image	**Product 360 image type**	**Product 360 image quality**	**Description**

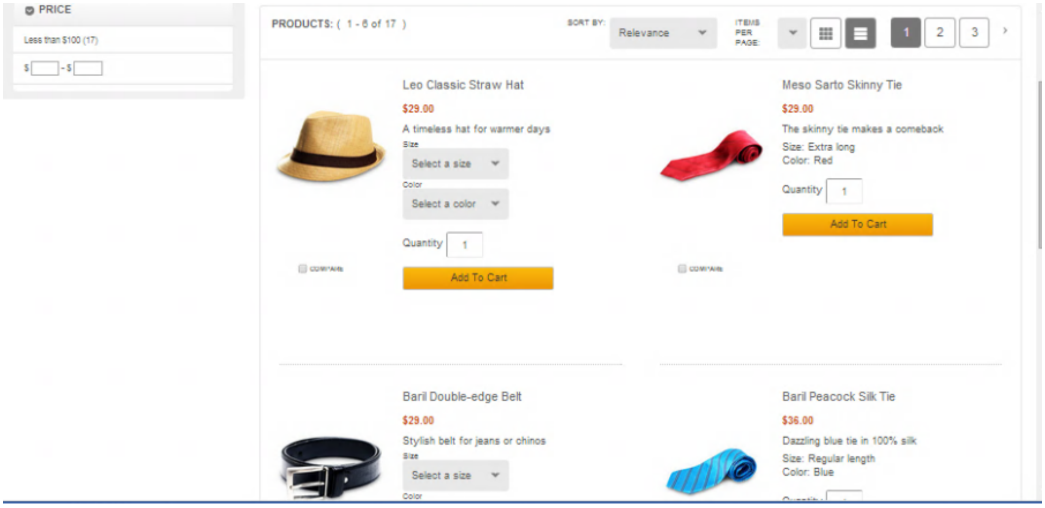
Full image	Standard image	Internet image	Provide a full image for products and/or items using the Product 360 image type *Standard image* and the Product 360 image quality *Internet image*. The visualization in the Madisons starter store sample is optimized for images of the size 160px × 160px.
Thumbnail	Thumbnail		Provide a thumbnail image for products and/or items using the Product 360 image type *Thumbnail* and the Product 360 image quality *Internet image*. The visualization in the Aurora starter store sample is optimized for images of the size 70px × 70px.

5.3.2 Catalog Entries Prices

5.3.2.1 Prices in WebSphere Commerce

WebSphere Commerce supports two types of prices; a list price and the offer prices.
The list price is directly assigned to the catalog entry and unique per currency. There is no quantity dependent scaling and no time period of validity which can be provided.
Offer prices are connected to a catalog entry through an offer. The offer allows specifying a start date end an end date. This can be used to specify time periods for the validity.
Additionally a minimum quantity can be specified by an offer, the offer price attached is valid from. Through this mechanism order quantity dependent sliding scale prices can be provided.

Hint: The Aurora starter store sample implementation shows issues with time periods and sliding scale prices when using product-item relations. Therefore it is not recommended to use this type of prices. Single items being not connected to a product seem to work properly in that area.



Leo Classic Straw Hat
$29.00
A timeless hat for warmer days
Size
Color
Quantity 1

Meso Sarto Skinny Tie
$29.00
The skinny tie makes a comeback
Size: Extra long
Color: Red
Quantity 1

Baril Double-edge Belt
$29.00
Stylish belt for jeans or chinos
Size
Color

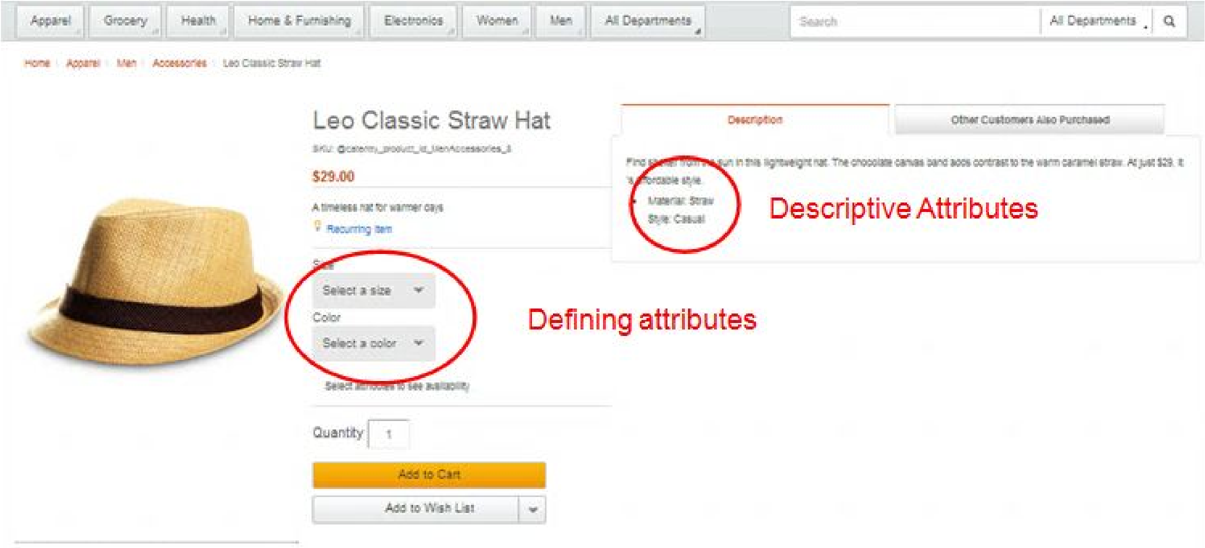
Baril Peacock Silk Tie
$36.00
Dazzling blue tie in 100% silk
Size: Regular length
Color: Blue

In the Aurora store implementation of WebSphere Commerce the list price is optional. An offer price has to be provided.
Without an offer price no price information is displayed, no putting to shopping cart and no ordering is possible even a list price would be available.
Simply the offer price is shown.

5.3.3 Attributes

5.3.3.1 Attributes in WebSphere Commerce

WebSphere Commerce allows having attributes on product and/or item level. There are two types of attributes known, the *descriptive attributes* and the *defining attributes*.



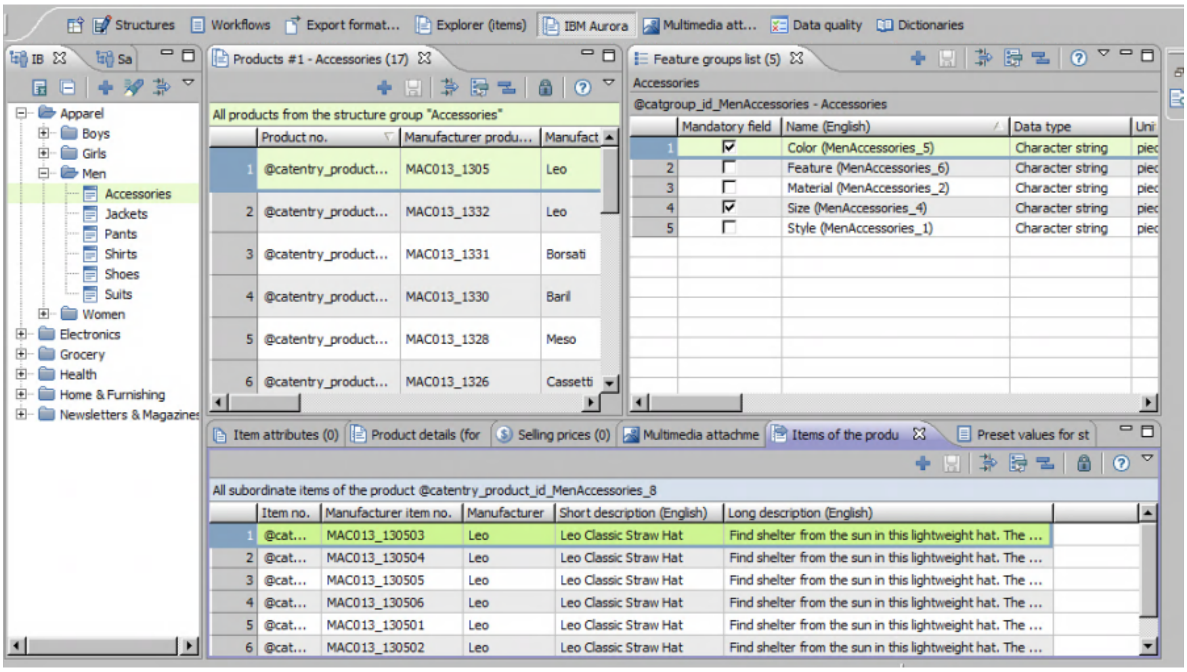
Leo Classic Straw Hat
SKU: @catentry_product_id_MenAccessories_1
$29.00
A timeless hat for warmer days
Size
Color
Select attributes to see availability
Quantity 1

Find shelter from the sun in this lightweight hat. The chocolate canvas band adds contrast to the warm caramel straw. At just $29, it 's affordable style.
- Material: Straw
- Style: Casual

Descriptive attributes are for information and display reasons only. They provide values and information which is valid for all variants of the products.
In the Aurora Starter Store implementation they are displayed for products on the bottom of its long description. Descriptive attributes of items are not displayed at all even they're supported by WebSphere Commerce in general.
Defining attributes on product level provide a selection of allowed values for each of the defining attributes. Each unique combination of values represents a variant of the product available.
Through the selection of a value for each of the defining attributes the item to order (the variant) is concretely specified – in this area the term *SKU resolution* is also familiar.

5.3.3.2 Attributes in Product 360

Product 360 in general distinguishes between the so called features and the attributes. The features are a kind of attribute template declared on structure group level.



All products from the structure group "Accessories"

	Product no.	Manufacturer produ...	Manufact
1	@catentry_product...	MAC013_1305	Leo
2	@catentry_product...	MAC013_1332	Leo
3	@catentry_product...	MAC013_1331	Borsati
4	@catentry_product...	MAC013_1330	Baril
5	@catentry_product...	MAC013_1328	Meso
6	@catentry_product...	MAC013_1326	Cassetti

Accessories

@catgroup_id_MenAccessories - Accessories

	Mandatory field	Name (English)	Data type	Uni
1	☑	Color (MenAccessories_5)	Character string	piec
2	□	Feature (MenAccessories_6)	Character string	piec
3	□	Material (MenAccessories_2)	Character string	piec
4	☑	Size (MenAccessories_4)	Character string	piec
5	□	Style (MenAccessories_1)	Character string	piec

All subordinate items of the product @catentry_product_id_MenAccessories_8

	Item no.	Manufacturer item no.	Manufacturer	Short description (English)	Long description (English)
1	@cat...	MAC013_130503	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
2	@cat...	MAC013_130504	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
3	@cat...	MAC013_130505	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
4	@cat...	MAC013_130506	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
5	@cat...	MAC013_130501	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...
6	@cat...	MAC013_130502	Leo	Leo Classic Straw Hat	Find shelter from the sun in this lightweight hat. The ...

Attributes belong to the products or items and provide a value for the structure group's feature the attribute is assigned to.
Attributes declaration starts on structure group level, where a feature list is provided specifying the attribute templates with the name, the data type and unit, etc.
The *Attribute type* signalizes the usage of the feature for products (*Product attribute*) or for items (*Item attribute*). Features of the attribute type *All* are used for both.
If a product is assigned to a structure group attributes for the features of the type *Product attribute* or *All* are created automatically.
If an item is assigned to a structure group attributes for the features of the type *Item attribute* or *All* are created automatically.
If a product with items already assigned will be added to a structure group the attributes on product and item level are created accordingly the types specified automatically.

> **Hint**: The usage of the correct Attribute type setting is helpful for efficient data maintenance on HPM side. It is not responsible for the usage of the attribute at the IBM WebSphere interface.

To control the usage of the attributes for WebSphere Commerce the right purpose has to be set. There are two purposes defining and descriptive which correspond with the meaning of that term in WebSphere Commerce.
Due descriptive attributes are only displayed on product level in the Aurora Starter Store implementation it makes sense to use this purpose in combination with the attribute type Product attribute and maintain values on product level only. Descriptive attributes are optional.
For defining attributes the products do not provide any value in Product 360. Specific values for the defining attributes are provided by the items representing a specific characteristic. Therefore it makes sense to combine the purpose Defining with the attribute type Item attribute.

Purpose	**Attribute type**	
Defining	Item attribute	Must. Each item of a product has to provide a value for each defining attribute. The set of values over all definig attributes has to be unique within the variants of the product.
Descriptive	Product attribute	Optional. As shown above descriptive attributes are displayed on product level in the Madisons store implementation.
Defining	Product attribute	No.
Descriptive	Item attribute	No, descriptive attributes on item level are not displayed in the Madisons store implementation.

The WebSphere Commerce system knows defining attributes on product level where the allowed values are maintained and stored. In HPM all defining attributes declared in the feature list of a structure group are exported as defining attributes tor each product of that group.
The list of allowed values provided for these defining attributes on product level are calculated from the distinct values of the corresponding defining attribute of the items assigned to the product automatically.
It has to be ensured that

- Every item has a value for each of the defining attributes
- The combination of values for all defining attributes has to be unique, specifying a unique variant of the product.
- Combinations not provided would lead to a message in the shops front-end when trying to add this combination to shopping cart.

For each item each defining attribute with the item specific value is exported additionally.

> **Hint:** All products and their items do share the same defining attributes as long they belong to the same structure group.

Some rules for proper features and attributes maintenance:
It is absolutely essential to have the feature list on group level properly defined:

- Ensure all feature do have a name in all languages transferred to WebSphere commerce.
- Ensure to have the *Purpose* of each feature defined well as *Descriptive* or *Defining*. Feature which should not be considered for WebSphere Commerce may not have one of these two purposes.
- Maintain the Attribute type maintained properly. Usually all descriptive features do have the attribute type Product attribute all defining features do have the attribute type Item attribute.
- Maintain the *Data type* for each feature properly. The data types being able to convert to the WebSphere Commerce types are Character string, Integer, Decimal, Floating point and Date. All other data types or if none is given will be interpreted as STRING by the data transfer.
- Maintain the field Sequence of the feature which will be used and interpreted with in the group's features of the same purpose.

A product will get attributes generated for each of the group's descriptive features (as long they are declared as Product attributes) when the product is assigned to the group.
These attributes will have the same names than the features and each of the attributes will be assigned to one of the features (see data field Assignment).

- To have this assignment in place is essential. If it got lost it has to be rebuilt by using the functionality called *Cumulative feature transfer* or manually by drag the products attribute onto the feature it has to be assigned to.
- Maintain a value for each of that descriptive attributes for each language which is transferred to WebSphere Commerce.
- For the data transfer only the value is taken from the attribute; all other data is taken from the group's feature.

An item will get attributes generated for each of the group's defining features (as long they are declared as Item attributes) when the item is assigned to the group or the product the item belongs to is assigned to the group.
These attributes will have the same names than the features and each of the attributes will be assigned to one of the features (see data field Assignment).

- To have this assignment in place is essential. If it got lost it has to be rebuilt by using the functionality called *Cumulative feature transfer* or manually by drag the products attribute onto the feature it has to be assigned to.
- Maintain a value for each of that defining attributes for each language which is transferred to WebSphere Commerce.
- For the data transfer only the value is taken from the attribute; all other data is taken from the group's feature.

> **Hint**: It is absolutely essential that all items do have the complete set of defining attributes declared for the group and specify a value for these attributes in all languages.
> The evaluation of the defining attributes per product and the allowed values list needed for WebSphere Commerce depends on this.
> Standalone items – not belonging to a product – may not have defining attributes. Therefore they may not be provided within structure groups where defining attributes are declared.

5.4 Attribute Dictionary

An IBM WebSphere Commerce attribute dictionary is a set of attributes and attribute values. The attribute dictionary can contain attributes with predefined values and attributes with assigned values. Predefined value attributes get a set of values that define the list of allowed values for the respective attribute. Assigned value attributes don't define such an allowed values list, assigned values are specified individually for each catalog entry.
Attributes and attribute values are independent from any catalog entries. You can assign attributes from the attribute dictionary to catalog entries (items and products) for use as defining attributes or descriptive attributes.

In WSC, all attributes of the attribute dictionary have a unique identifier.In order to meet the need we do the following:

In Product 360 we have two levels of (structure/ structure group) attributes:

- The pool (each pool feature has a unique identifier)
- Structure group level (one pool feature can be used at several structure groups with different data types, preset values, ...)

We combine data of the two levels to create as few as possible unique attributes. So one pool feature - many structure group attributes - some attributes for the attribute dictionary

The following algorithm is used to collect data for the attribute dictionary:

- Collect all structure group attributes.
- Group them by identifier (it's the identifier of the pool feature) and data type.
- Cumulate the preset values of each attribute group.

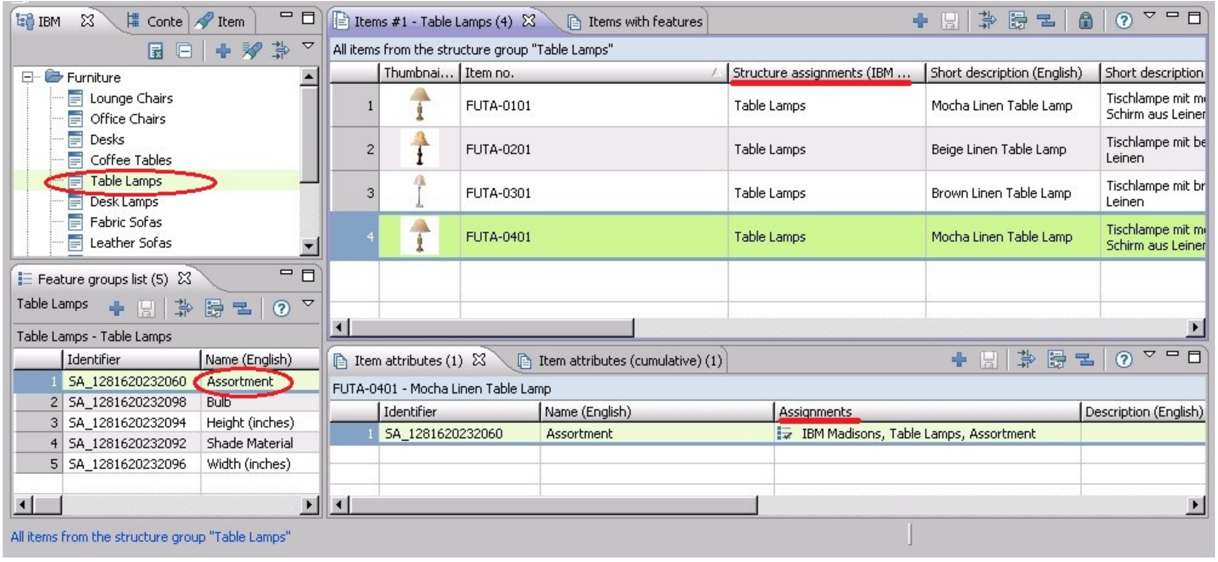
So each group will become an attribute of the dictionary, the identifier is *<structure attribute identifier>_<id of the data type>*, e.g. **size_1** (the "size" feature with `string` data type), **size_4** (the "size" feature with `integer` data type).

Please refer to Attribute Dictionary for further details.

To make sure that a structure group feature will be exported, ensure following steps:

- the structure group feature has to be mapped to a structure feature

- the structure group feature has to be mapped to at least one item and/or product. Free attributes are not exported.



All items from the structure group "Table Lamps"

	Thumbnai...	Item no.	Structure assignments (IBM ...	Short description (English)	Short description
1		FUTA-0101	Table Lamps	Mocha Linen Table Lamp	Tischlampe mit m Schirm aus Leiner
2		FUTA-0201	Table Lamps	Beige Linen Table Lamp	Tischlampe mit be Leinen
3		FUTA-0301	Table Lamps	Brown Linen Table Lamp	Tischlampe mit br Leinen
4		FUTA-0401	Table Lamps	Mocha Linen Table Lamp	Tischlampe mit m Schirm aus Leiner

Table Lamps - Table Lamps

	Identifier	Name (English)
1	SA_1281620232060	Assortment
2	SA_1281620232098	Bulb
3	SA_1281620232094	Height (inches)
4	SA_1281620232092	Shade Material
5	SA_1281620232096	Width (inches)

FUTA-0401 - Mocha Linen Table Lamp

	Identifier	Name (English)	Assignments	Description (English)
1	SA_1281620232060	Assortment	IBM Madisons, Table Lamps, Assortment	

All items from the structure group "Table Lamps"

- the attribute needs to have a valid value.

The preset values of a structure group feature will be accumulated for each attribute group, hence there may be not a single entry per preset value.

5.4.1 Computing the list of dictionary attributes

We only use the structure attribute identifier and the data type of the structure group attributes to compute the list of attributes for the attribute dictionary. We ignore other data that might be necessary in other scenarios, like units.

5.4.2 Multi value support

The multi value support for describing catalog entry attributes is disabled to avoid attribute value cumulation.

5.4.3 Product 360 field - WCS field mapping

Some WCS fields are filled with constant values because there're no matching standard fields in Product 360. For custom solutions it is possible to use appropriate repository reserve fields.

5.4.3.1 Features without preset values

The value will be only available in one language for features without preset values. This is the actual last exported language entry of one specific attribute dictionary value in the CSV file.

5.4.3.2 Item/Product attributes

Free attributes won't get exported, only mapped attributes will be gathered.

5.4.4 Deleted allowed values, deleted attributes

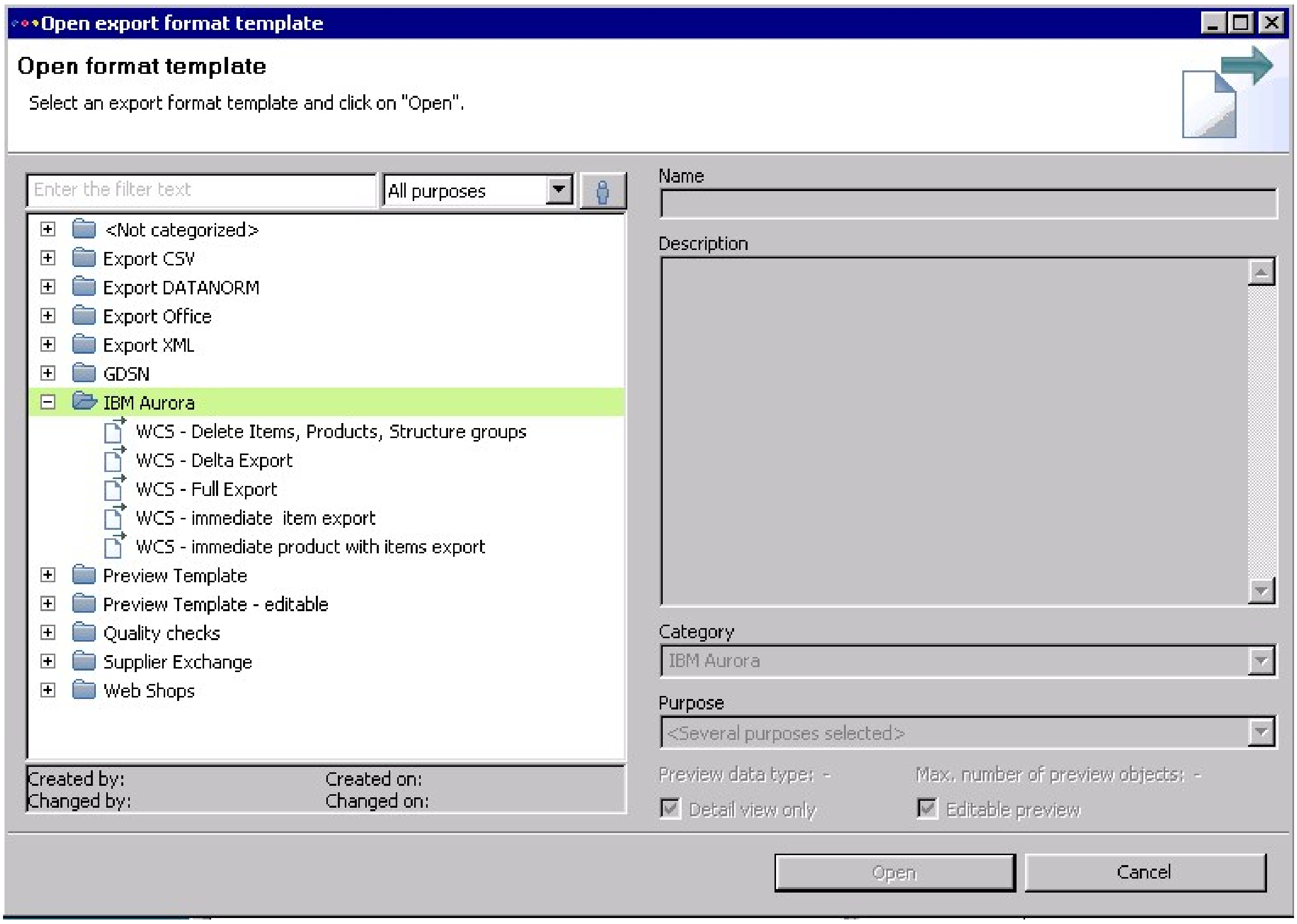
We don't support the export of deleted attributes or attribute allowed values respectively attribute dictionary entries and attribute dictionary values entries.

5.5 Exports

Exports out of Product 360 to WebSphere Commerce can be triggered the following ways:

1. Quick Update
2. Delta Export
3. Full Export.

The following export templates are provided to meet the above functionalities:



Open export format template

Open format template

Select an export format template and click on "Open".

Enter the filter text | All purposes

- ⊞ <Not categorized>
- ⊞ Export CSV
- ⊞ Export DATANORM
- ⊞ Export Office
- ⊞ Export XML
- ⊞ GDSN
- ⊟ IBM Aurora
  - WCS - Delete Items, Products, Structure groups
  - WCS - Delta Export
  - WCS - Full Export
  - WCS - immediate item export
  - WCS - immediate product with items export
- ⊞ Preview Template
- ⊞ Preview Template - editable
- ⊞ Quality checks
- ⊞ Supplier Exchange
- ⊞ Web Shops

Created by: Created on:
Changed by: Changed on:

Name

Description

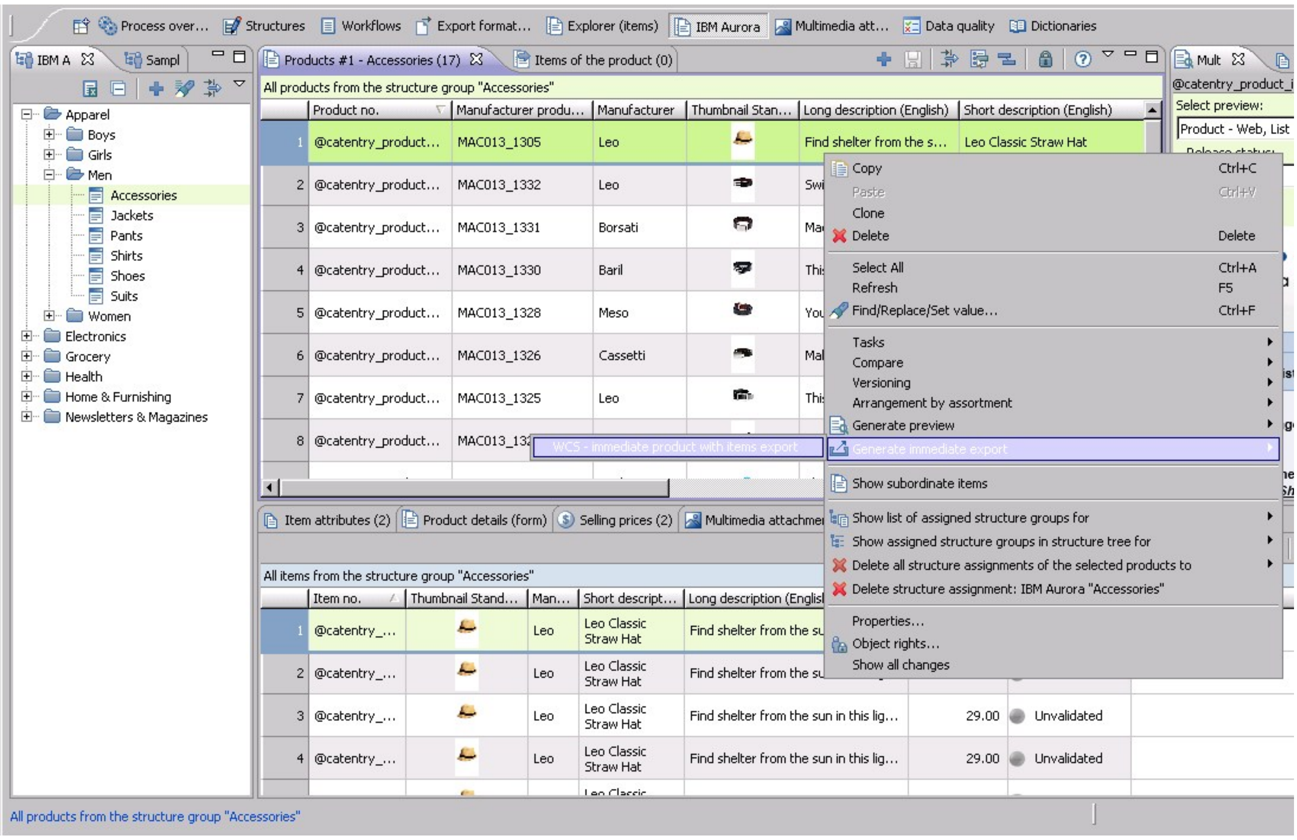
Category: IBM Aurora

Purpose: <Several purposes selected>

Preview data type: - Max. number of preview objects: -

☑ Detail view only ☑ Editable preview

For immediate export right click on a product or an item, go to "Generate Immediate Export" then click on "WCS-Immediate product with item Export" to trigger the job to WCS as shown the picture below



All products from the structure group "Accessories"

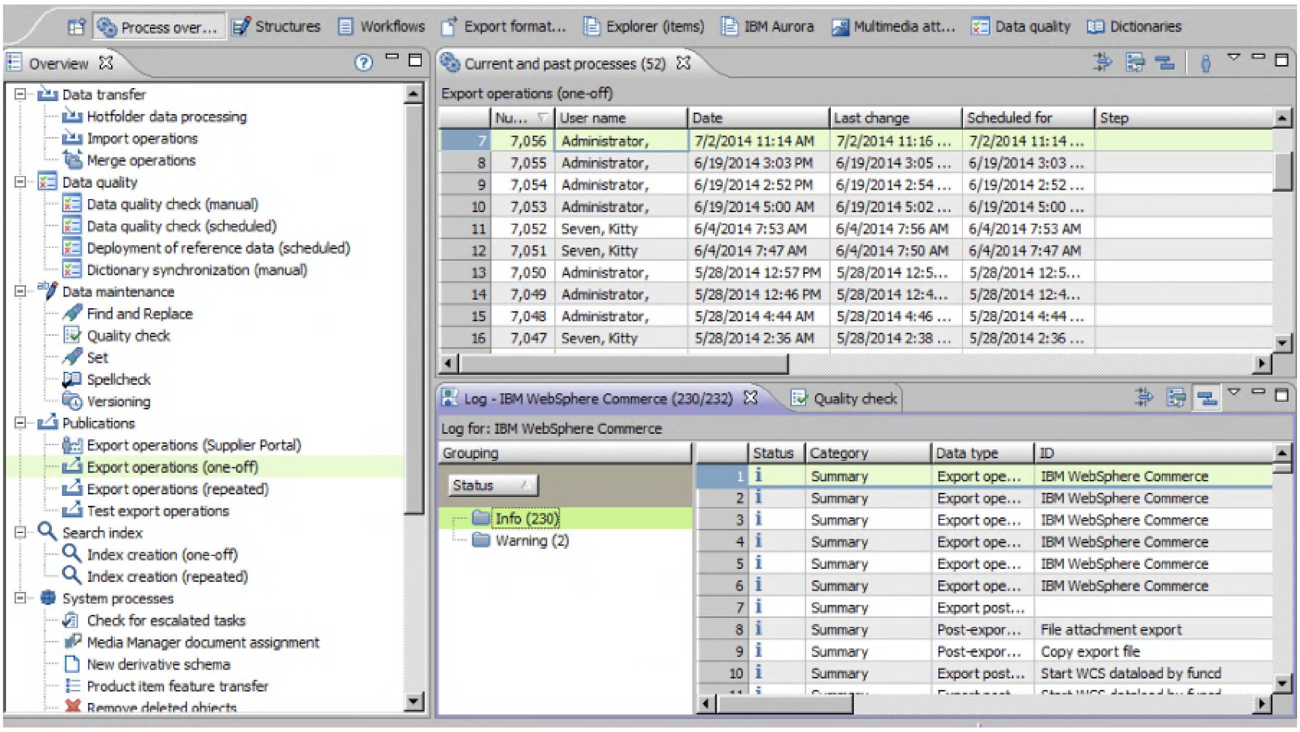
	Product no.	Manufacturer produ...	Manufacturer	Thumbnail Stan...	Long description (English)	Short description (English)
1	@catentry_product...	MAC013_1305	Leo		Find shelter from the s...	Leo Classic Straw Hat
2	@catentry_product...	MAC013_1332	Leo		Swi	
3	@catentry_product...	MAC013_1331	Borsati		Ma	
4	@catentry_product...	MAC013_1330	Baril		Thi	
5	@catentry_product...	MAC013_1328	Meso		You	
6	@catentry_product...	MAC013_1326	Cassetti		Mal	
7	@catentry_product...	MAC013_1325	Leo		Thi	
8	@catentry_product...	MAC013_13				

All items from the structure group "Accessories"

	Item no.	Thumbnail Stand...	Man...	Short descript...	Long description (Englis		
1	@catentry_...		Leo	Leo Classic Straw Hat	Find shelter from the su		
2	@catentry_...		Leo	Leo Classic Straw Hat	Find shelter from the su		
3	@catentry_...		Leo	Leo Classic Straw Hat	Find shelter from the sun in this lig...	29.00	Unvalidated
4	@catentry_...		Leo	Leo Classic Straw Hat	Find shelter from the sun in this lig...	29.00	Unvalidated
				Leo Classic			

All products from the structure group "Accessories"

To monitor the progress please go to the "Process Overview Perspective" to check for status of the job.



- Data transfer
  - Hotfolder data processing
  - Import operations
  - Merge operations
- Data quality
  - Data quality check (manual)
  - Data quality check (scheduled)
  - Deployment of reference data (scheduled)
  - Dictionary synchronization (manual)
- Data maintenance
  - Find and Replace
  - Quality check
  - Set
  - Spellcheck
  - Versioning
- Publications
  - Export operations (Supplier Portal)
  - Export operations (one-off)
  - Export operations (repeated)
  - Test export operations
- Search index
  - Index creation (one-off)
  - Index creation (repeated)
- System processes
  - Check for escalated tasks
  - Media Manager document assignment
  - New derivative schema
  - Product item feature transfer
  - Remove deleted objects

Export operations (one-off)

	Nu...	User name	Date	Last change	Scheduled for	Step
7	7,056	Administrator,	7/2/2014 11:14 AM	7/2/2014 11:16 ...	7/2/2014 11:14 ...	
8	7,055	Administrator,	6/19/2014 3:03 PM	6/19/2014 3:05 ...	6/19/2014 3:03 ...	
9	7,054	Administrator,	6/19/2014 2:52 PM	6/19/2014 2:54 ...	6/19/2014 2:52 ...	
10	7,053	Administrator,	6/19/2014 5:00 AM	6/19/2014 5:02 ...	6/19/2014 5:00 ...	
11	7,052	Seven, Kitty	6/4/2014 7:53 AM	6/4/2014 7:56 AM	6/4/2014 7:53 AM	
12	7,051	Seven, Kitty	6/4/2014 7:47 AM	6/4/2014 7:50 AM	6/4/2014 7:47 AM	
13	7,050	Administrator,	5/28/2014 12:57 PM	5/28/2014 12:5...	5/28/2014 12:5...	
14	7,049	Administrator,	5/28/2014 12:46 PM	5/28/2014 12:4...	5/28/2014 12:4...	
15	7,048	Administrator,	5/28/2014 4:44 AM	5/28/2014 4:46 ...	5/28/2014 4:44 ...	
16	7,047	Seven, Kitty	5/28/2014 2:36 AM	5/28/2014 2:38 ...	5/28/2014 2:36 ...	

Log for: IBM WebSphere Commerce

Grouping

Status

- Info (230)
- Warning (2)

	Status	Category	Data type	ID
1	i	Summary	Export ope...	IBM WebSphere Commerce
2	i	Summary	Export ope...	IBM WebSphere Commerce
3	i	Summary	Export ope...	IBM WebSphere Commerce
4	i	Summary	Export ope...	IBM WebSphere Commerce
5	i	Summary	Export ope...	IBM WebSphere Commerce
6	i	Summary	Export ope...	IBM WebSphere Commerce
7	i	Summary	Export post...	
8	i	Summary	Post-expor...	File attachment export
9	i	Summary	Post-expor...	Copy export file
10	i	Summary	Export post...	Start WCS dataload by funcd

5.5.1 Limitations

5.5.1.1 Exporting into the master catalog of IBM WebSphere

Adding a new category to the master catalog does not automatically add that category to the visible IBM WebSphere store. The final step is to link your new master catalog category to the sales catalog for the Aurora esite store. You can find the steps described here: http://www.ibm.com/support/knowledgecenter/SSZLC2_8.0.0/com.ibm.commerce.management-center.doc/tasks/tpnlinkctgctlog.htm

5.5.1.2 Descriptive products attributes without preset values

Currently only one language of the exported attribute values will be imported in the WebSphere Commerce system. There is actually no workaround for this.

5.5.1.3 Delta Export

Changes to Product/Item association types and referenced number is not supported. The workaround is to delete the reference and create a new one.

Moving structure groups to another parent is not supported. The workaround is to create a new structure and reassign the Products/Items.

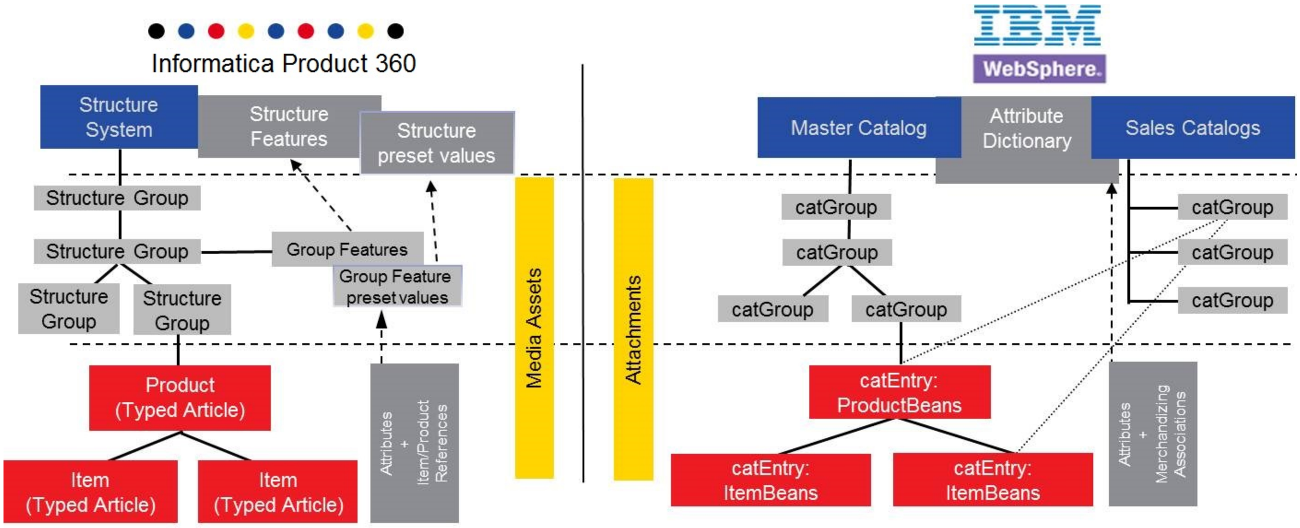
5.5.1.4 Immediate Export

The immediate Export format templates will never export deleted data. Furthermore the export will fail if the following Objects are not already on the WCS system:

- The new mapped attribute values must be already in the WebSphere Commerce system.
- The Structure group of the Product/Item must be in the WebSphere Commerce system.
- The parent product of an item must already be there if an item is being exported.

6 Field mapping

6.1 Overview



On the image above you can see some differences of the Informatica Product 360 model compared to the IBM WebSphere Commerce model. This page is describing how the Product 360 fields got mapped to the according WCS fields.

6.2 Formatting and mapping for export format templates

6.2.1 Numbers

Numbers will be formatted by the `FormatDecimal` export function.

Example: `{?FormatDecimal {&Selling price.Price (from 1)}, ".", 2}`

6.2.2 Dates

Dates will be formatted by the `FormatDate` export function.

Example: `{?FormatDate {&Selling prices (with price tiers).Valid from}, "yyyy-MM-dd 00:00:00.000000"}`

6.2.3 Language

The language id mapping is resolved by the `GetWCSLanguageId` export function.

Example: `{?GetWCSLanguageId {&Language-specific data.Language}}`

All supported languages are listed in the Appendix B: Language Mapping(see page 93). If you miss a language mapping you can add it to the plugin_customization.ini file as follows:

plugin_customization.ini - language mapping

```
# pattern: hpm.lang.<internal language id>=<WCS language id>
# Dutch
com.heiler.ppm.export.extension.ibm.core/hpm.lang.19=-17
```

6.2.4 Data type

The data type mapping is resolved by the `GetWCSDatatype` export function.

Example: `{?GetWCSDatatype {&Attribute dictionary attributes.Data type}}`

The supported data types are listed in the Appendix C: Data types Mapping(see page 94).

plugin_customization.ini - data type mapping

```
# pattern: hpm.datatype.<key of the Enum.Datatypes enumeration>=<WCS data type
identifier>
# Integer
com.heiler.ppm.export.extension.ibm.core/hpm.datatype.4=INTEGER
```

6.2.5 Association type

The association type mapping is resolved by the `GetWCSItemReferenceType` export function.

Example: `{?GetWCSItemReferenceType {&Referenced products.Reference type}}`

Find the association type mapping in the Appendix D: Item Reference Types Mapping(see page 94).

plugin_customization.ini - association type mapping

```
# pattern: hpm.reftype.<key of the Enum.ArticleReferenceType enumeration>=<WCS
association identifier>
# accessories
com.heiler.ppm.export.extension.ibm.core/hpm.reftype.6=ACCESSORY
```

6.2.6 Sequence

The default value of sequences in Product 360 is 2147483647 which means "the sequence is not maintained yet", the corresponding default value for WCS is 0.

That's why the output of sequences should be adjusted using the `GetWCSDisplaySequence` export function.

Example: `{?GetWCSDisplaySequence{&ADAttributeRelations.Sequence}}`

6.2.7 Strings

Names, descriptions, etc. have to be escaped by the `EscapeValue` export function.

Example: `{?EscapeValue {&Language-specific data.Name}}`

6.2.8 Keywords

Keywords have to be separated by commas and included in quotation marks.

The output of keywords has to be adjusted using the `Replace` export function, all semicolons have to be replaced by commas.

Example: `{?EscapeValue {?Replace {&Language-specific data.Keywords},";",","}}`

6.2.9 Images

There're two information of images that need to be exported: the image itself and the path under which the image is stored in the zip file.

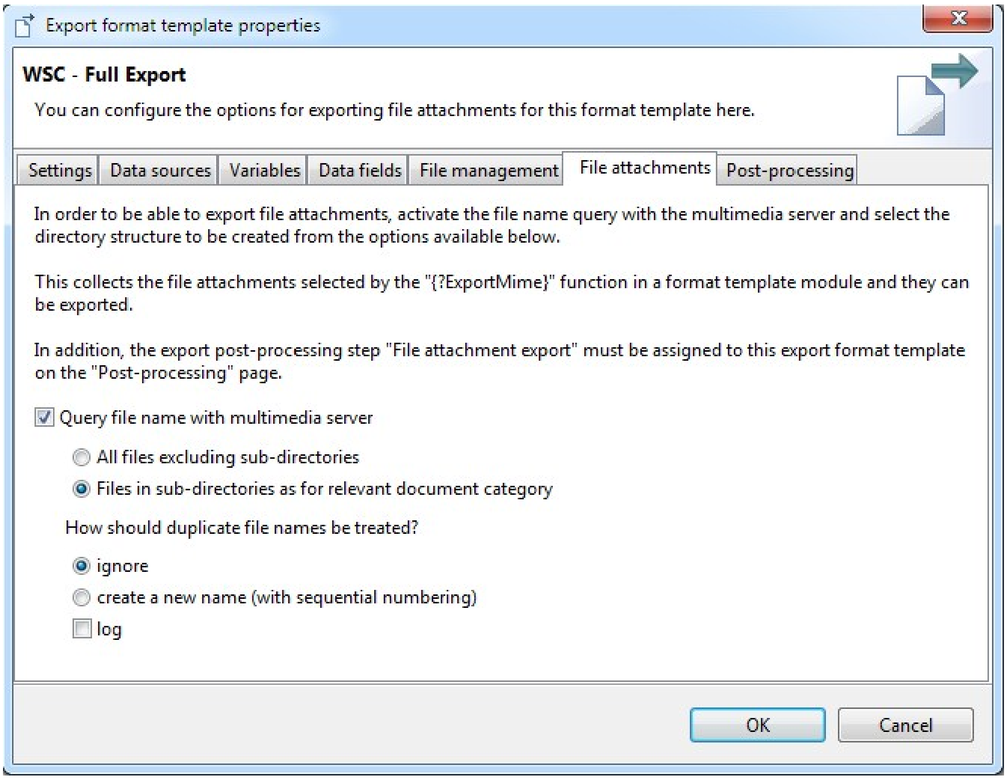
The `ExportMime` export function is used to mark the image as to be exported and outputs the relative path of the respective file.

All images (full images, thumbnails of structure groups, products, items) are exported like this:

```
{%IMAGE_PATH_RELATIVE}{?ExportMime {&Document identifier}, {&Quality}}
```

Note: the image path depends on the respective Product 360 system, use the IMAGE_PATH_RELATIVE variable to adapt the resulting path with a prefix, e.g. "images/catalog/".

Ensure the *File attachment* settings of the export template's properties are set correctly. See the following screen shot.



Export format template properties

WSC - Full Export

You can configure the options for exporting file attachments for this format template here.

Settings | Data sources | Variables | Data fields | File management | File attachments | Post-processing

In order to be able to export file attachments, activate the file name query with the multimedia server and select the directory structure to be created from the options available below.

This collects the file attachments selected by the "{?ExportMime}" function in a format template module and they can be exported.

In addition, the export post-processing step "File attachment export" must be assigned to this export format template on the "Post-processing" page.

☑ Query file name with multimedia server

○ All files excluding sub-directories

● Files in sub-directories as for relevant document category

How should duplicate file names be treated?

● ignore

○ create a new name (with sequential numbering)

□ log

6.3 Data consistency

6.3.1 Data field length

Most of the WebSphere Commerce text containing data base fields in Oracle are defined as varchar2 (nnn BYTE). Due UTF-8 text coming from Product 360 can have 1 to 3 bytes per character this text has to be checked and trimmed to the byte length by the "May not be longer than the maximum number of bytes" data field validation rule without damaging the UTF-8 characters.

6.4 The Attribute Dictionary data exchange

An attribute dictionary is a set of common attributes and attribute values that can be reused by multiple products.

6.4.1 Attribute Dictionary - Attributes

Contains attribute dictionary attributes. These attributes can be used for descriptive or SKU resolution purposes, and can be shared by catalog entries.

6.4.1.1 Template Configuration

- Export data provider: Structure
- Export data type: Attribute dictionary attributes

Data type filter

- Purpose: <all purposes>
- Language: "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Output file

- ADAttributes.csv

6.4.1.2 Sample for an attribute dictionary attributes' import file

ADAttributes.csv

```
AttributeIdentifier|Type|AttributeType|LanguageId|Name|Description|Sequence|
Displayable|Searchable|Comparable|Merchandisable|Delete
SF_COLOR_1|STRING|AllowedValues|-1|"Color"|||true|true|true|true|0
SF_COLOR_1|STRING|AllowedValues|-3|"Farbe"|||true|true|true|true|0
SF_SIZE_1|STRING|AllowedValues|-3|"Größe"|||true|true|true|true|0
...
```

6.4.1.3 Field mapping

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Attribute Identifier	ATTR. IDENTIFIER	VARCHAR (254) NOT NULL	An external identifier for the attribute.	computed field: Attribute dictionary attributes.Dictionary attribute identifier (StructureFeature. Identifier + "_" + StructureGroupAttribute. Datatype)	String, 64	Special field of "Attribute dictionary attributes" export sub-data type

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Type	ATTR. ATTRTYPE_ID	CHAR (16) NOT NULL	The attribute type. For example, INTEGER, STRING, or FLOAT.	Attribute dictionary attributes.Data type (StructureGroup Attribute. Datatype)	Long	
Attribute Type	ATTR. ATTRUSAGE	INTEGER	Defines the usage of the attribute. 1 - Attribute with allowed values. When assign this type of attribute to catalog entry, user need to choose from the predefined allowed values. 2 or null - Attribute with assigned values.This type of attribute does not have predefined allowed value. When assign this type of attribute to catalog entry, user need to input a value. This value belongs to the catalog entry.	Attribute dictionary attributes.Has preset values	Boolean	Special field of "Attribute dictionary attributes" export sub-data type, evaluates if the cumulated preset values list is empty. • yes = "AssignedValues" • no = "AllowedValues"
LanguageId	ATTRDESC. LANGUAGE_ID	INTEGER NOT NULL	The language for this attribute description.	Attribute dictionary attributes.Language of the name (StructureGroup AttributeLang. LanguageID)	Long	
Name	ATTRDESC. NAME	VARCHAR (254) NOT NULL	The name of the attribute.	Attribute dictionary attributes.Name (StructureGroup AttributeLang. Name)	String, 200	

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Description	ATTRDESC. DESCRIPTION	VARCHAR (254)	The attribute description.	Attribute dictionary attributes.Description (StructureGroupAttributeLang. Description)	String, unlimited	The data field length is checked and will be truncated with warning if necessary.
Sequence	ATTR. SEQUENCE	DOUBLE NOT NULL DEFAULT 0	The display order of attributes in an attribute group or in the root of the attribute dictionary.	Attribute dictionary attributes.Sequence (StructureGroupAttribute. DisplayOrder)	Integer, unlimited	
Displayable	ATTR. DISPLAYABLE	INTEGER	This flag identifies if this attribute is displayable on the store front. This flag can be used to temporarily hide an attribute on the store front. It can also used for attributes that belong to a web crawler (such as Google's web crawling robot), that is not for display to customer. A value of 1 or null indicates that the attribute is displayable, while a value of 0 indicates that the attribute is not displayable.	static value "true" (after Repository configuration StructureFeature. Displayable)		

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Searchable	ATTR. SEARCHABLE	INTEGER	This flag identifies if this attribute can be searched. Searchable attributes can be indexed to search engine. A value of 1 indicates that the attribute is searchable, while a value of 0 or null indicates that the attribute is not searchable.	static value "true" (after Repository configuration StructureFeature. Searchable)		
Comparable	ATTR. COMPARABLE	INTEGER	This flag identifies if this attribute can be used for comparison. A value of 1 indicates that the attribute can be used for comparison, while a value of 0 or null indicates that the attribute cannot be used for comparison. For example, attribute "Auto On/Off" is marked as comparable for all coffee makers. If a shopper selects this attribute on the store front page, only coffee makers that have the "Auto On/Off" feature will be displayed.	static value "true" (after Repository configuration StructureFeature. Comparable)		

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Merchandisable	ATTR.MERCHANDISABLE	INTEGER	Specifies that the attribute is used in creating merchandising rules. This indicates that the attribute is for internal business use only and does not display to customers unless it is also set to be displayable, searchable, or faceable.	static value "true" (after Repository configuration StructureFeature.Merchandisable)		
Delete				static value 0		For insert/update = 0, to delete = 1

6.4.2 Attribute Dictionary – Allowed Values

The attribute values that can be shared globally by different attributes of different catalog entries.

6.4.2.1 Template Configuration

- Export data provider: Structure
- Export data type: Attribute dictionary attribute values

Data type filter

- Purpose: <all purposes>
- Language: "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Output file

- ADAttributeAllowedValues.csv

6.4.2.2 Sample for an 'attribute dictionary attributes allowed values' import file

ADAttributeAllowedValues.csv

```
AttributeIdentifier|ValueIdentifier|Sequence|Value|ValueUsage|Image1|Image2|
LanguageId|Delete
SF_COLOR_1|SV_RED||"rot"|1|||-3|0
SF_COLOR_1|SV_RED||"red"|1|||-1|0
SF_COLOR_1|SV_BLUE||"blau"|1|||-3|0
SF_COLOR_1|SV_BLUE||"blue"|1|||-1|0
...
SF_SIZE_1|SV_SIZE_S||"S"|1|||-1|0
SF_SIZE_1|SV_SIZE_S||"S"|1|||-3|0
SF_SIZE_1|SV_SIZE_M||"M"|1|||-1|0
SF_SIZE_1|SV_SIZE_M||"M"|1|||-3|0
...
```

6.4.2.3 Field mapping

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Attribute Identifier	ATTRVAL. ATTR_ID	BIGINT NOT NULL	The external identifier for the attribute. Which is resolved to the internal unique identifier (ATTR.ATTR_ID) for this attribute through table ATTR.	computed field: Attribute dictionary attributes.Dictionary attribute identifier (StructureFeature. Identifier + "_" + StructureGroupAttribute. Datatype)	String, 64	Special field of "Attribute dictionary attribute values" export sub-data type
ValueIdentifier	ATTRVAL. IDENTIFIER	VARCHAR (254)	An external identifier for the attribute value.	Attribute dictionary attribute values.Preset value.Identifier (StructureValue. Identifier)	String, 60	

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Sequence	ATTRVALDESC. SEQUENCE	DOUBLE	A number that determines the display order of a list of allowable attribute values for a given attribute.	Attribute dictionary attribute values.Preset value sequence (StructureGroupAttributePresetValue. DisplayOrder)	Integer, unlimited	
Value	ATTRVALDESC. STRINGVALUE or INTEGERVALUE or FLOATVALUE	VARCHAR (2000) or INTEGER or DOUBLE	If this type of attribute value is STRING, then this column will hold the attribute value. If the type is not STRING, then this column will be NULL. If this type of attribute value is INTEGER, then this column will hold the attribute value. If the type is not INTEGER, then this column will be NULL. If this type of attribute value is FLOAT, then this column will hold the attribute value. If the type is not FLOAT, then this column will be NULL.	Attribute dictionary attribute values.Name (StructureValue. Name)	String, 2000	

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
ValueUsage	ATTRVAL. VALUSAGE	INTEGER	**1** Predefined allowed value. (For example, one of the allowed values of Color is Blue) **2** Default predefined allowed value (When assign attribute to catalog entry, default predefined allowed value will be selected by default) **null** Assigned attribute value. (For example Product's Length is 30. Length attribute does not have allowed values).	Static value '1' (after Repository configuration StructureValue .Usage)		Always part of the allowed value list.
Image1	ATTRVALDESC. IMAGE1	VARCHAR (254)	The path of the first image of this attribute value.	(empty)		Not supported
Image2	ATTRVALDESC. IMAGE2	VARCHAR (254)	The path of the second image of this attribute value.	(empty)		Not supported
LanguageId	ATTRVALDESC. LANGUAGE_ID		The language for this attribute value description.	Attribute dictionary attribute values.Language (StructureValueLang. LanguageID)		

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Delete						For insert/ update = 0, to delete = 1

6.5 The catalog groups data exchange

One of the structure systems maintained in Product 360 can be used as the master catalog structure to be transferred to WCS.

6.5.1 Catalog Groups

With the catalog groups file the most basic information on catalog groups is provided; the unique group identifier and the information if a certain group is a top group. Top level groups have to be marked with TRUE in the column TopGroup. Non top level groups have to have the value FALSE for this column.

6.5.1.1 Template Configuration

- Export data provider: Structure groups (All groups in structure system (with product filter))
- Export data type: Structure group

Data provider parameter

- Structure system: the structure system specified for "structure system" data provider
- Product assortment (filter): none
  That means all structure groups are exported, irrespective of whether there're mapped to products or not

Sorting

Structure group.Level: Catalog groups have to be sorted by their levels because we first need the parent groups and then the children

Output file

- CatGroups.csv

6.5.1.2 Sample for a catalog groups' import file

CatGroups.csv

```
GroupIdentifier|ParentGroupIdentifier|TopGroup|Sequence|Name|LanguageId|Delete
SG_MEN||TRUE|1001|SG_MEN|-1|0
...
SG_WOMEN_SHIRTS|SG_WOMEN|FALSE|10|SG_WOMEN_SHIRTS|-1|0
SG_MEN_SHOES|SG_MEN|FALSE|3|SG_MEN_SHOES|-1|0
SG_WOMEN_PANTS|SG_WOMEN|FALSE|20|SG_WOMEN_PANTS|-1|0
SG_MEN_PANTS|SG_MEN|FALSE|2|SG_MEN_PANTS|-1|0
...
```

6.5.1.3 Field mapping

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
GroupIdentifier	CATGROUP. IDENTIFIER	VARCHAR2 (254 BYTE)	The external name that is used to identify the catalog group. Along with MEMBER_ID, these columns are a unique index.	Structure group. Structure group identifier	String, 60	The unique identifier of the group.
ParentGroupIdentifier	CATGRPREL .CATGROUP_ID_PARENT	NUMBER NOT NULL [VARCHAR2 (254 BYTE)]	The source catalog group of this relationship.	Structure group. Parent identifier	String, 60	The unique group identifier of the desired parent group. Must stay empty for the top level groups.
TopGroup				static value "TRUE" or "FALSE" (depends on Structure group.Level)		"TRUE" for top level groups, "FALSE" for all children (any level > 1)

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Sequence	CATTOGRP. SEQUENCE resp. CATGRPREL. SEQUENCE	DOUBLE resp. DOUBLE NOT NULL (DEFAULT 0)	The display sequence for top level groups. resp. The display sequence for none top level groups.	Structure group. Sequence	Integer	May stay empty.
Name	CATGRPDESC. NAME	VARCHAR2 (254 BYTE) NOT NULL	The language-dependent name of this catalog group.	Structure group.Structure group identifier	String, 60	The unique identifier is used as dummy field. The structure group names will be imported by the Catalog Groups Description(see page 71).
LanguageId	CATGRPDESC. LANGUAGE_ID	NUMBER NOT NULL	The identifier of the language. For a list of language components, see the LANGUAGE table.	Variable: defaultLanguage (by default English)	Long	The variable must be equals to the WC shop default language.
Delete	CATGROUP. MARKFORDELETE	INTEGER NOT NULL	Indicates if this catalog group has been marked for deletion: 0 = No, 1 = Yes		Timestamp	For insert/ update = 0, to delete = 1

6.5.2 Catalog Groups Description

With the catalog groups description file the language dependent text information plus image paths are transported.

6.5.2.1 Template Configuration

- Export data provider: Structure groups (All groups in structure system (with product filter))
- Export data type: Language-specific data

Data type filter

- Language = "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Output file

- CatGroupDescription.csv

6.5.2.2 Collecting image data

There're two sub-modules in the export templates for collecting both image types, full images and thumbnails.

Full image data

Template Configuration

- Export data provider: Structure groups (All groups in structure system (with product filter))
- Export data type: Structure group file attachments

Data type filter

- Quality = "Internet image" (Variable FULLIMAGE_QUALITY)
- Type = "Standard image" (Variable FULLIMAGE_TYPE)
- Language = "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Thumbnail data

Template Configuration

- Export data provider: Structure groups (All groups in structure system (with product filter))
- Export data type: Structure group file attachments

Data type filter

- Quality = "Internet image" (Variable THUMBNAIL_QUALITY)
- Type = "Thumbnail" (Variable THUMBNAIL_TYPE)
- Language = "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Field mapping

CSV file column	Product 360 data field	Remarks

FullImage	{%IMAGE_PATH_RELATIVE}
	{?ExportMime {&Document identifier},
	{&Quality}}
		See Images

CSV file column	Product 360 data field	Remarks
LanguageId	Language	

6.5.2.3 Sample for a catalog groups' description import file

CatGroupsDescription.csv

```
GroupIdentifier|LanguageId|Name|ShortDescription|LongDescription|Published|Keyword|
Note|FullImage|Thumbnail|Delete
...
SG_MEN_SHIRTS|-1|"Shirts"|"Short description: Shirts"|"Long description: Shirts"|1|||
images/IMG_0019_e.jpg|images/IMG_0019_e.jpg|0
SG_MEN_SHIRTS|-3|"Hemden"|"Kurzbeschreibung: Hemden"|"Langbeschreibung: Hemden"|1|||
images/IMG_0028_e.jpg|images/IMG_0028_e.jpg|0
SG_WOMEN_SHIRTS|-1|"Shirts"|"Short description: Shirts"|"Long description: Shirts"|1|
||images/IMG_0117_e.jpg|images/IMG_0117_e.jpg|0
...
```

6.5.2.4 Field mapping

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
GroupIdentifier	CATGRPDESC. CATGROUP_ID	NUMBER NOT NULL [VARCHAR2 (254 BYTE)]	The internal reference number of the catalog group.	Structure group. Structure group identifier	String, 60	The unique identifier provided before is resolved to the corresponding internal id.
LanguageId	CATGRPDESC. LANGUAGE_ID	NUMBER NOT NULL	The identifier of the language. For a list of language components, see the LANGUAGE table.	Structure group. Language	Long	

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Name	CATGRPDESC. NAME	VARCHAR2 (254 BYTE) NOT NULL	The language-dependent name of this catalog group.	Structure group. Name	String, 250	
ShortDescription	CATGRPDESC. SHORTDESCRIPTION	VARCHAR2 (254 BYTE)	A short description of this catalog group.	Structure group. Description	String, unlimited	The data field length is checked and will be truncated with warning if necessary.
LongDescription	CATGRPDESC. LONGDESCRIPTION	VARCHAR2 (4000 BYTE)	A long description of this catalog group.	Structure group. Comment	String, unlimited	The data field length is checked and will be truncated with warning if necessary.
Published	CATGRPDESC. PUBLISHED	NUMBER NOT NULL	Indicates whether this catalog group should be displayed for the language.	static value "1" (after Repository configuration Structure group .Published)		Publish = 1 Don't publish = 0
Keyword	CATGRPDESC. KEYWORD	VARCHAR2 (254 BYTE)	A keyword used for searching.	Structure group.Keywords	String, 250	The data field length is checked and will be truncated with warning if necessary.
Note	CATGRPDESC. NOTE	VARCHAR2 (4000 BYTE)	The footnotes or extended long description for this catalog group.	Not used (after Repository configuration Structure group.Note)		The sample implementation does not use and visualize this field.

CSV column name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
FullImage	CATGRPDESC. FULLIMAGE	VARCHAR2 (254 BYTE)	The full image path of this catalog group.	see Collecting image data	String, 2000	
Thumbnail	CATGRPDESC. THUMBNAIL	VARCHAR2 (254 BYTE)	The thumbnail image path of this catalog group.	see Collecting image data	String, 2000	
Delete						For insert/ update = 0,to delete = 1

6.6 The catalog entries data exchange

Catalog entries in the understanding of IBM WebSphere Commerce are represented by the logical data entities Product and Item of Informatica Product 360.

6.6.1 Catalog Entries

With the catalog entries' import file the basic information on catalog entries is provided; the unique part number and the object type (Product or Item) and its name in the defined language. In addition the mapping of the catalog entries (both products and items) to the master catalog groups is imported (ParentGroupIdentifier) and the unique part number of the parent catalog entry (the product) an item belongs to (ParentPartNumber). The field Buyable signals if the item is buyable in general.

6.6.1.1 Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: Product, Item

Output file

- CatEntries.csv

6.6.1.2 Sample for a catalog entries' import file

CatEntries.csv

```
PartNumber|Type|ParentGroupIdentifier|ParentPartNumber|Name|LanguageId|Buyable|Delete
P2_D|Product|SG_WOMEN_SHIRTS||P2_D|-1|0|0
P2_D|Product|SG_WOMEN_SHIRTS||P2_D|-3|0|0
...
A8|Item||P2_D|A8|-1|1|0
A8|Item||P2_D|A8|-3|1|0
A7|Item||P2_D|A7|-1|1|0
A7|Item||P2_D|A7|-3|1|0
...
```

6.6.1.3 Field mapping

CSV col. name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
PartNumber	CATENTRY. PARTNUMBER	VARCHAR2(64 BYTE) NOT NULL	The reference number that identifies the part number of the catalog entry. Along with the MEMBER_ID, these columns are a unique index.	Product. Product no., Items. Item no.	String, 250	The unique identifier for the catalog entry.
Type	CATENTRY. CATENTTYPE_ID	VARCHAR2(16 BYTE) NOT NULL	Identifies the type of catalog entry. Foreign key to the CATENTTYPE table. The supported default types are: ProductBean, ItemBean, PackageBean, BundleBean and DynamicKitBean.	static value "Product" or "Item"		Due data export for products and items in Product 360 is performed in separate modules fix values can be provided.

CSV col. name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
ParentGroupIdentifier	CATGPENREL. CATGROUP_ID	NUMBER NOT NULL [VARCHAR2(254 BYTE)]	The source catalog group of this relationship.	Product. Structure group. Structure group identifier	String, 250	The unique group identifier provided with catalog group before is resolved to the corresponding internal id. It's only filled for products.
ParentPartNumber	CATENTREL. CATENTRY_ID _PARENT	NUMBER NOT NULL [VARCHAR2(64 BYTE)]	The reference number of the source catalog entry in this relationship.	Higher-level product. Referenced product number	String, 250	The unique part identifier of the product is resolved to the corresponding internal id. It's only filled for items.
Name	CATENTDESC. NAME	VARCHAR2 (128 BYTE)	The language-dependent name of this catalog entry.	Product. Product no., Item. Object no.	String, 250	Dummy field, needed for WCS. Will be overwritten with the Product/ Item name during "CatalogEntryDescription" import
LanguageId	CATENTDESC. LANGUAGE_ID	NUMBER NOT NULL	The identifier of the language. For a list of language components, see the LANGUAGE table.	Product. Language-specific data . Language	Long	Dummy field, needed for WCS. Will be overwritten with the Product/ Item name during "CatalogEntryDescription" import

CSV col. name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
Buyable				static value for products = 0 (not buyable); for items = 1 (buyable) (after Repository configuration Product. Buyable, Item. Buyable)		
Delete						For insert/ update = 0,to delete = 1

6.6.2 Catalog Entries Description

The file for the catalog entries' descriptions transports the language dependent text information, keywords and images. The publication state of a catalog entry is also part of this import file.

6.6.2.1 Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: Language-specific data

> Please be aware that image paths will only be exported if there are descriptions in corresponding languages.

Data type filter

- Language: "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Output file

- CatEntryDescription.csv

6.6.2.2 Collecting image data

There're two sub-modules in the export templates for collecting the both image types, full images and thumbnails.

Full image data

Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: File attachments

Data type filter

- Quality = "Internet image" (Variable FULLIMAGE_QUALITY)
- Type = "Standard image" (Variable FULLIMAGE_TYPE)
- Language = "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Thumbnail data

Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: File attachments

Data type filter

- Quality = "Internet image" (Variable THUMBNAIL_QUALITY)
- Type = "Thumbnail" (Variable THUMBNAIL_TYPE)
- Language = "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Field mapping

CSV file column	Product 360 data field	Remarks
LanguageId	Language	
FullImage	{%IMAGE_PATH_RELATIVE}{?ExportMime {&Document identifier},{&Quality}}	See Images(see page 59)

6.6.2.3 Sample for a catalog entries' descriptions import file

CatEntriesDescription.csv

```
PartNumber|Type|LanguageId|Name|ShortDescription|LongDescription|AuxDescription1|
AuxDescription2|Keyword|Published|FullImage|Thumbnail|Delete
0001.01|Product|-1|"Fitted Shirt"|"Lightweight casual shirt"|"The unique
fabric..."|||"shirt"|1|images/IMG_0101_f.jpg|images/IMG_0101_t.jpg|0
0001.01|Product|-3|"Tailliertes Hemd"|"Hemd aus leichtem Stoff"|"Eine einmaligen
Qualität..."|||"Hemd"|1|images/IMG_0101_f.jpg|images/IMG_0101_t.jpg|0
...
0099.02|Item|-1|"Dress Shirt"|"Tailored Dress Shirt"|"This dress shirt
features..."|||"Shirt"|1|images/IMG_9902_f.jpg|images/IMG_9902_t.jpg|0
0099.02|Item|-3|"Anzughemd"|"Tailliertes Anzughemd"|"Dieses Anzughemd..."|||"Anzughem
d"|1|images/IMG_9902_f2.jpg|images/IMG_9902_t2.jpg|0
...
```

6.6.2.4 Field mapping

CSV Col. name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
PartNumber	CATENTDESC. CATENTRY_ID	NUMBER NOT NULL [VARCHAR2 (64 BYTE)]	The target catalog entry of this relationship.	Product.Product no., Item.Item no.	String, 250	The unique part identifier provided with catalog entry before is resolved to the corresponding internal id.
Type	CATENTRY. CATENTTYPE_ID	VARCHAR2(16 BYTE) NOT NULL	Identifies the type of catalog entry. Foreign key to the CATENTTYPE table. The supported default types are: ProductBean, ItemBean, PackageBean, BundleBean and DynamicKitBean.	static value "Product" or "Item"		Due data export for products and items in Product 360 is performed in separate modules fix values can be provided.

CSV Col. name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
LanguageId	CATENTDESC. LANGUAGE_ID	NUMBER NOT NULL	The identifier of the language. For a list of language components, see the LANGUAGE table.	Product.Language, Item.Language	Long	
Name	CATENTDESC. NAME	VARCHAR2 (128 BYTE)	The language-dependent name of this catalog entry.	Product.Short description, Item.Short description	String, 254	The data field length is checked and will be truncated with warning if necessary.
ShortDescription	CATENTDESC. SHORTDESCRIPTION	VARCHAR2 (254 BYTE)	A short description of this catalog entry.	Products.Other remarks, Item.Other remarks	String, 254	The data field length is checked and will be truncated with warning if necessary.
LongDescription	CATENTDESC. LONGDESCRIPTION	CLOB	A long description of this catalog entry.	Products.Long description, Item.Long description	String, unlimited	
AuxDescription1	CATENTDESC. AUXDESCRIPTION1	VARCHAR2 (4000 BYTE)	Additional description for this catalog entry.	(after Repository configuration on Product. AuxDescription1, Item.AuxDescription1)		

CSV Col. name	WC Col. Name	WC Col. Type	WC Description	Product 360 data field	Product 360 data type	Note
AuxDescription 2	CATENTDESC. AUXDESCRIPTION2	VARCHAR2 (4000 BYTE)	Additional description for this catalog entry.	(after Repository configuration Product. AuxDescription2, Item. AuxDescription2)		
Keyword	CATENTDESC. KEYWORD	VARCHAR2 (254 BYTE)	A keyword used for searching.	Products.Keywords, Item .Keywords	String, 50	The data field length is checked and will be truncated with warning if necessary.
Published	CATENTDESC. PUBLISHED	NUMBER NOT NULL	Indicates whether this catalog entry should be displayed for the language indicated by LANGUAGE_ID: 0= catalog entry should not be displayed, 1= catalog entry should be displayed.	static value "1" (after Repository configuration Product.Published, Item.Published)		
FullImage	CATENTDESC. FULLIMAGE	VARCHAR2 (254 BYTE)	The full image path of this catalog entry.	see Collecting image data(see page 0)	String, 2000	
Thumbnail	CATENTDESC. THUMBNAIL	VARCHAR2 (254 BYTE)	The thumbnail image path of this catalog entry.	see Collecting image data(see page 0)	String, 2000	
Delete						For insert/ update = 0,to delete = 1

6.6.3 Catalog Entries' Attribute Values

Contains the catalog entry and attribute dictionary attribute relationship.

6.6.3.1 Template Configuration

- Export data provider: Products, Items
- Export data type: Attribute dictionary attribute relations

Data type filter

- Structure system: Variable "MASTERCATALOG_STRUCTURE"
- Purpose: <all purposes>
- Language: "English" (Variable defaultLanguage), "German" (Variable secondaryLanguage)

Data field validation

- Field: ArticleAttributeValue.Value
  Validation: May not be blank
  Action on error: remove data record
  Error classification: warning

Output file

- CatEntryADAttributeValues.csv

6.6.3.2 Sample for a catalog entries' attribute values import file

CatEntryADAttributeValues.csv

```
PartNumber|Type|AttributeIdentifier|LanguageId|ValueIdentifier|Value|Usage|Sequence|
Delete
0001.01|Product|SF_CARE_INSTRUCTION_1|-3||"waschen"|Descriptive||0
0001.01|Product|SF_CARE_INSTRUCTION_1|-1||"n/a"|Descriptive||0
0001.01|Product|SF_MADE_IN_GERMANY_1|-3|SV_YES||Descriptive||0
0001.01|Product|SF_MADE_IN_GERMANY_1|-1|SV_YES||Descriptive||0
0001.01|Product|SF_MADE_IN_CHINA_1|-3|SV_NO||Descriptive||0
0001.01|Product|SF_MADE_IN_CHINA_1|-1|SV_NO||Descriptive||0
...
```

6.6.3.3 Field mapping

CSV column name	WC Column Name	WC Column Type	WC Description	Product 360 data field	Product 360 data type	Note

PartNumber	CATENTRYATTR.CATENTRY_ID	BIGINT NOT NULL	The logical identifier of the catalog entry. Resolved to a unique catalog entry id.	Product.Product no., Item.Item no.	String, 250	The unique part identifier provided with catalog entry.
Type			The object type of the catalog entry (product or item).	static value "Product" or "Item"		
Attribute Identifier	CATENTRYATTR.ATTR_ID	BIGINT NOT NULL	The logical identifier of the attribute. Resolved through the attributes' dictionary (ATTR) to a unique attribute id.	computed field: ADAttributeRelations. Dictionary attribute identifier (StructureFeature. Identifier + "_" + StructureGroupAttribute. Datatype)	String, 64	Special field of "Attribute dictionary attribute relations" export sub-data type
LanguageId				ADAttributeRelations. Language of value (ProductAttributeValue. LanguageID, ArticleAttributeValue. LanguageID)	Long	
ValueIdentifier	CATENTRYATTR.ATTRVAL_ID	BIGINT NOT NULL DEFAULT 0	The reference number to an attribute value in the ATTRVAL table. The value 0 means the attribute is assigned to the catalog entry without a specific value.	ADAttributeRelations. Structure preset value identifier (StructureValue. Identifier)		The identifier of the assigned preset value in case there is a preset value.

Value				ADAttributeRelations. Attribute value (ProductAttributeValue. Value, ArticleAttributeValue. Value)	String, 2000	Empty if there is a ValueIdentifier.

Usage	CATENTRYATTR. USAGE	CHAR (1) NOT NULL	'1' for defining attribute (sku resolution)'2' for descriptive attribute.	for Items: ADAttributeRelations. Purpose (ArticleAttribute. Purpose)
				for Products: static value "Descriptive"
					String, 1000	Note: The defining attribute purpose has to be defined in the repository enumeration "Enum.AttributePurpose" as enum param with the name "Websphere.defining", current value is "DEFINING"

Sequence	CATENTRYATTR. SEQUENCE	DOUBLE	A number that determines the display order of an attribute for a given catalog entry.	ADAttributeRelations. Sequence (ProductAttribute. DisplayOrder, ArticleAttribute. DisplayOrder)	Integer, unlimited	
Delete						

6.6.4 Catalog Entries' Associations

With the catalog entries' merchandising associations import file the marketing oriented mappings between catalog entries (both products and items) are imported.

6.6.4.1 Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: Referenced products, Referenced items

Data type filter

- Reference type = Accessories, Cross selling, Successor item, Up selling
- Product assortment = ProductAssortment (Variable)

- Item assortment = ItemAssortment (Variable)

Output file

- CatEntryAssociation.csv

6.6.4.2 Sample for a catalog entries' associations import file

CatEntryAssociation.csv

```
PartNumber|AssociationType|TargetPartNumber|Sequence|Quantity|Delete
P11|UPSELL|P13|1|1|0
P3_D|X-SELL|P1_D|1|1|0
P12|UPSELL|P13|1|1|0
P5|X-SELL|P1_D|1|1|0
A19|ACCESSORY|A1|1|2|0
...
```

6.6.4.3 Field mapping

CSV column name	WC Column Name	WC Column Type	WC Description	HPM data field	HPM data type	Note
PartNumber	MASSOCCECE.CATENTRY_ID_FROM	NUMBER NOT NULL (VARCHAR (64) NOT NULL)	The CatalogEntry that is the source of the association.	Product. Product no., Item. Item no.	String, 250	The unique part identifier provided with catalog entry before is resolved to the corresponding internal id.
AssociationType	MASSOCCECE.MASSOCTYPE_ID	VARCHAR2 (32 BYTE) NOT NULL	The Identifier of the association type. Foreign key to the MASSOCTYPE table (allowed values are:X-SELL, UPSELL, ACCESSORY, REPLACEMENT).	Referenced products. Reference type, Referenced items. Reference type	Converted to allowed value	The HPM item reference identifier is converted to the WC identifier by export function `GetWCSItemReferenceType`. See Appendix D: Item Reference Types Mapping

CSV column name	WC Column Name	WC Column Type	WC Description	HPM data field	HPM data type	Note
TargetPartNumber	MASSOCCECE. CATENTRY_ID_TO	NUMBER NOT NULL (VARCHAR (64) NOT NULL)	The CatalogEntry that is the target of the association.	Referenced products. Referenced object number, Referenced items. Referenced object no.	String, 250	The unique part identifier of the referenced part is resolved to the corresponding internal id.
Sequence	MASSOCCECE. RANK	NUMBER(20,5)	The sequence number used for display order.	Referenced products. Sequence, Referenced items. Sequence	Long	Default is "1"
Quantity	MASSOCCECE. QUANTITY	NUMBER	The quantity related to this association.	Referenced products. Number, Referenced items. Number	Integer	The number of items referenced.
Delete						For insert/ update = 0,to delete = 1

6.6.5 Catalog Entries Prices - list prices

Prices of catalog entries are imported in two portions, the list price and the offer price(s) for each catalog entry.

Several prices in different currencies may be provided per catalog entry.

6.6.5.1 Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: Selling price

Data type filter

- Currency = "US Dollar" (Variable LISTPRICE_CURRENCY)
- Customer = "<Public>" (Variable LISTPRICE_CUSTOMER)
- Price type = "Non-binding price recommendation" (Variable LISTPRICE_PRICETYPE)

- Time of validity = TODAY (dynamic variable)
- Valid in region = "USA" (Variable LISTPRICE_TERRITORY)

Output file

- CatEntryListPrices.csv

6.6.5.2 Sample for a catalog entries' list price import file:

CatEntryListPrices.csv

```
PriceListName|PartNumber|Price|CurrencyCode|Delete
AuroraList|P9|99.99|USD|0
AuroraList|P4|99.99|USD|0
...
```

6.6.5.3 Field mapping

CSV column name	WC Column Name	WC Column Type	WC Description	Product 360 data field	Product 360 data type	Note
PriceListName			The name of the price list the price belongs to.	variable		Variable PRICELIST_NAME, by default "AuroraList"
PartNumber	LISTPRICE.CATENTRY_ID	NUMBER NOT NULL	The target catalog entry of this relationship.	Product. Product no, Item. Item no.	String, 250	The unique part identifier provided with catalog entry before is resolved to the corresponding internal id.
Price	LISTPRICE.LISTPRICE	NUMBER(20,5) NOT NULL	The amount of the ListPrice.	Selling price. Price (from 1)	BigDecimal	Formatted by export function FormatDecimal using "." as decimal separator and two fraction digits.
PriceCurrency	LISTPRICE.CURRENCY	VARCHAR2(3 BYTE) NOT NULL	The currency of the prices.This is a currency code as per ISO 4217 standards.	Selling price. Currency	String, 3	Provided as 3-digit ISO code by export function IsoCodeCurrency.

Delete						For insert/update = 0,to delete = 1

6.6.6 Catalog Entries Prices - offer prices

Prices of catalog entries are imported in two portions, the list price and the offer price(s) for each catalog entry.

Prices for several currencies may be provided per catalog entry.

6.6.6.1 Template Configuration

- Export data provider: Product list, Items by product assignments
- Export data type: Selling price (with price tiers)

Data type filter

- Currency = "US Dollar" (Variable OFFERPRICE_CURRENCY)
- Customer = "<Public>" (Variable OFFERPRICE_CUSTOMER)
- Price type = "Net customer price" (Variable OFFERPRICE_PRICETYPE)
- Time of validity = TODAY (dynamic variable, contains the respective current date)
- Valid in region = "USA" (Variable OFFERPRICE_TERRITORY)

Output file

- CatEntryListPrices.csv

6.6.6.2 Sample for a catalog entries' offer price import file

CatEntryOfferPrices.csv

```
PriceListName|PartNumber|StartDate|EndDate|Price|CurrencyCode|MinimumQuantity|Delete
Aurora|P9|2014-04-02 00:00:00.000000|9999-12-31 23:59:59.999999|45.00|USD|1.00|0
Aurora|P4|2014-04-02 00:00:00.000000|9999-12-31 23:59:59.999999|67.00|USD|1.00|0
Aurora|P13|2014-04-02 00:00:00.000000|9999-12-31 23:59:59.999999|33.00|USD|1.00|0
...
```

6.6.6.3 Field mapping

CSV column name	WC Column Name	WC Column Type	WC Description	Product 360 data field	Product 360 data type	Note
PriceListName			The name of the price list the price belongs to.	Variable		Variable PRICELIST_NAME, by default "AuroraList"
PartNumber	OFFER. CATENTRY_ID	NUMBER NOT NULL [VARCHAR2 (64 BYTE)]	The target catalog entry of this relationship.	Product. product no., Item. item no.	String, 250	The unique part identifier provided with catalog entry before is resolved to the corresponding internal id.
StartDate	OFFER. STARTDATE	TIMESTAMP(6)	The price for the nominal quantity (see CATENTSHIP. NOMINALQUANTITY) of the product referred to by the Offer.	Selling prices (with price tiers). Valid from	Date	Formated as "yyyy-MM-dd 00:00:00.000000"
EndDate	OFFER. ENDDATE	TIMESTAMP(6)	The currency of the prices.This is a currency code as per ISO 4217 standards.	Selling prices (with price tiers). Valid until	Date	Formatted as "yyyy-MM-dd 23:59:59.999999"
Price	OFFERPRICE. PRICE	NUMBER(20,5) NOT NULL	The price for the nominal quantity (see CATENTSHIP. NOMINALQUANTITY) of the product referred to by the Offer.	Selling prices (with price tiers). Price (from 1)	BigDecimal	Formatted by export function FormatDecimal using "." as decimal separator and two fraction digits.

CSV column name	WC Column Name	WC Column Type	WC Description	Product 360 data field	Product 360 data type	Note
CurrencyCode	OFFERPRICE. CURRENCY	VARCHAR2(3 BYTE) NOT NULL	The currency of the prices.This is a currency code as per ISO 4217 standards.	Selling prices (with price tiers). Currency	String, 3	Provided as 3-digit ISO code by export function IsoCodeCurrency.
MinimumQuantity	OFFER. MINIMUMQUANTITY	NUMBER	The minimum quantity that can be purchased in a single Order under this Offer.	Selling prices (with price tiers). Lower price tier	BigDecimal	Formatted using "." as decimal separator and two fraction digits.
Delete						For insert/ update = 0,to delete = 1

6.7 Appendix A: Variables of the Export Templates

The variables ensure values used at different places of the export template are the same at each usage. The value can be maintained at a central place at the export template's setting.
Please adapt the values to your Product 360 – WC – System environment.

Variable	Default value	Description
Images		
IMAGE_PATH_RELATIVE		
FULLIMAGE_QUALITY	Internet image	The Product 360 image quality to be used for the full image of products, items and structure groups.
FULLIMAGE_TYPE	Standard image	The Product 360 image type to be used for the full image of products, items and structure groups.
THUMBNAIL_QUALITY	Internet image	The Product 360 image quality to be used for the thumbnail image of products, items and structure groups.
THUMBNAIL_TYPE	Thumbnail	The Product 360 image type to be used for the thumbnail image of products, items and structure groups.

Prices		
LISTPRICE_PRICETYPE	Non-binding price recommendation	The Product 360 price type used for the list price.
LISTPRICE_TERRITORY	USA	The Product 360 validity region the list price is valid for. Product 360 needs for price access, there is no meaning for WC.
LISTPRICE_CUSTOMER	<Public>	The Product 360 customer the list price is valid for. Product 360 needs for price access, there is no meaning for WC.
LISTPRICE_CURRENCY	US Dollar	
PRICELIST_NAME	AuroraList	Name for price list of list prices
TODAY	Always evaluated the current date	The current date to evaluate the valid list price for.
OFFERPRICE_PRICETYPE	Net customer price	The Product 360 price type used for the offer price.
OFFERPRICE_TERRITORY	USA	The Product 360 validity region the offer price is valid for. Product 360 needs for price access, there is no meaning for WC.
OFFERPRICE_CUSTOMER	<Public>	The Product 360 customer the offer price is valid for. Product 360 needs for price access, there is no meaning for WC.
OFFERPRICE_CURRENCY	US Dollar	
OFFERPRICELIST_NAME	Aurora	Name for price list of offer prices:
Different		
MASTERCATALOG_STRUCTURE	(To be defined before first usage)	The Product 360 Structure System to be used as master catalog for the Web Shop System. Select from the structure systems available in your Product 360.

DELIMITER	|	Defines the token delimiter to be used in exported files.
		The token delimiter can be configured in the WCS data loader configuration files ("wc-loader*.xml"):
		wc-loader-*.xml
		`<_config:DataReader className="com.ibm.commerce.foundation.dataload.datareader.CSVReader"`
		` tokenDelimiter="|"`
		`          firstLineIsHeader="true"`
		` useHeaderAsColumnName="true" />`

defaultLanguage	English	This is the language on WCS side. It's also used to export the language specific data of Product 360.
secondaryLanguage	German	The secondary is used, next to the defaultLanguage, for all language specific data.
exportMultimedia	Yes	Specifies if the "Multimedia export" post step should run
ProductAssortment	(To be defined before first usage)	Specifies the products to be exported; all products should be mapped to the MASTERCATALOG_STRUCTURE. In addition, the item assortment is computed by that given product assortment.
updateItemAssortment	Yes	The item assortment will be used to filter the item associations. In this way external associations won't get exported. This flag indicates if the item assortment(should be updated or not.
updateProductAssortment	Yes	Used to compute the new and changed products and assigned items. With this flag you can define if the product assortment should be updated or not.
DELTA_FROM_DATE	6/5/2014 12:56 PM (must be changed)	This date variable is used to define a date from which the delta should be taken.

6.8 Appendix B: Language Mapping

The language id mapping between Product 360 and WCS:

Language	Product 360	WCS
English (US)	9	-1

French	12	-2
German	7	-3
Dutch	19	-17
Russian	25	-20
Czech	5	-41
Slovak	27	-42

6.9 Appendix C: Data Types Mapping

The data types mapping between Product 360 and WCS:

Product 360 (Key, Synonym)	**Product 360Display Name(en)**	**Product 360Display Name (de)**	**WCS**
1, STRING	Character string	Zeichenkette	STRING
4, INTEGER	Integer	Ganzzahl	INTEGER
2, DECIMAL	Decimal	Dezimal	FLOAT
3, DOUBLE	Floating point	Fließkomma	FLOAT
6, DATETIME	Date	Datum	DATETIME
-	-	-	FREEFORM
-	-	-	BIGINT
-	-	-	ATTACHMENT
5, BOOLEAN	Logical value	Wahrheitswert	STRING

6.10 Appendix D: Item Reference Types Mapping

The item reference types mapping between Product 360 and WCS:

Product 360	Product 360 Display Name (en)	Product 360 Display Name (de)	WCS	WCS Description
10, xsell	Cross-selling	Cross-Selling	X-SELL	Cross-sell merchandising relationship
11, upsell	Up-selling	Up-Selling	UPSELL	Up-sell merchandising relationship
6, accessories	Accessories	Zubehör	ACCESSORY	Accessory merchandising relationship
3, followup	Successor item	Nachfolgeartikel	REPLACEMENT	Replacement relationship

6.11 Appendix E: Object Types Mapping

The object types mapping between Product 360 and WCS:

Product 360	WCS
Article	Item
Product	Product

6.12 Repository configuration

The standard Product 360 repository doesn't contain all necessary fields to match all available fields of IBM. Therefore the standard WCS export format templates are using static variables. To use all available fields instead of static variables are following repository changes necessary.

> Please note that this is a demo showcase which will use repository reserve fields.

6.12.1 Changes in Repository Types

Change used reserve fields from inactive to active

6.12.2 Changes in repository custom

Please set the new created fields to visible and also set the export purpose to (at least) 1. For further information about repository field type properties see Domain Model (Repository).

Field length

Please also adjust the max length of the fields to the limits which are supported by the IBM WebSphere Commerce system. Find more information on Field mapping.

Added fields to Structure Feature

Displayable	
Identifier	StructureFeature.WCS.Displayable
Field Type	StructureFeatureType.Res_Bit_01

Searchable	
Identifier	StructureFeature.WCS.Searchable
Field Type	StructureFeatureType.Res_Bit_02

Comparable	
Identifier	StructureFeature.WCS.Comparable
Field Type	StructureFeatureType.Res_Bit_03

Merchandisable	
Identifier	StructureFeature.WCS.Merchandisable
Field Type	StructureFeatureType.Res_Bit_04

6.12.3 Added fields to Structure Value

Usage	
Identifier	StructureValue.WCS.Usage
Field Type	StructureValueType.Res_Bit_01
Lower Bound	1
Upper Bound	1
Value	1

6.12.3.1 Added fields to Structure Group

Published	
Identifier	StructureGroup.WCS.Published
Field Type	StructureGroupType.Res_Bit_01
Lower Bound	1
Upper Bound	1
Value	1

Note	
Identifier	StructureGroup.WCS.Note
Field Type	StructureGroupType.Res_Text2G_01
Max Length	4000

6.12.3.2 Added fields to Item

Buyable	
Identifier	Article.WCS.Buyable
Field Type	ArticleType.Res_Bit_01
Lower Bound	1
Upper Bound	1

Buyable	
Value	1

Published	
Identifier	Identifier: Article.WCS.Published
Field Type	ArticleType.Res_Bit_02
Lower Bound	1
Upper Bound	1
Value	1

Added fields to Item language specific data

AuxDescription1	
Identifier	ArticleLang.WCS.AuxDescription1
Field Type	ArticleLangType.Res_Text2G_01
Max Length	4000

AuxDescription2	
Identifier	ArticleLang.WCS.AuxDescription2
Field Type	ArticleLangType.Res_Text2G_02
Max Length	4000

6.12.3.3 Added fields to Product

Buyable	
Identifier	Product2G.WCS.Buyable
Field Type	ArticleType.Res_Bit_01
Lower Bound	1
Upper Bound	1
Value	0

Published	
Identifier	Product2G.WCS.Published
Field Type	ArticleType.Res_Bit_02
Lower Bound	1
Upper Bound	1
Value	1

Added fields to Product language specific data

AuxDescription1	
Identifier	Product2GLang.WCS.AuxDescription1
Field Type	ArticleLangType.Res_Text2G_01
Max Length	4000
AuxDescription2	
Identifier	Product2GLang.WCS.AuxDescription2
Field Type	ArticleLangType.Res_Text2G_02
Max Length	4000

7 Sample data

The IBM WebSphere Commerce Accelerator is delivered with sample data to show the functionality.

7.1 How to add sample data

The following steps and import files are only usable with the classic provider

1. Create maintenance structure with identifier "`WCS`"
2. Import all import projects
    - Adjust "Structure" parameter value for following import mappings and fields if structure identifier is not "WCS":
        i. 5_StructureGroupFeatures: "`Feature name (key)`"
        ii. 6_StructureGroupFeaturePresetValues: "`Feature name (key)`"
        iii. 7_BaseData: "`Structure`"
        iv. 12_FeatureValues: "`Structure`"
3. Unzip the Multimedia.zip file in the "`filestorage.mime.path`" which is defined in the server.properties