

StockLevels Import Half Interface

Technical Design Documentation

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updated By | Date | Comments |
| 0.1 | LEVY Quentin | 02/03/2015 | Initial Draft |
|  |  |  |  |
|  |  |  |  |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature/Date |
| Pascal ESPINOUSE |  |  |
| RECOUVREUR FLAVIEN |  |  |
| DURANT DE LA PASTELLIERE BENOIT |  |  |

Table of Contents

[1. Introduction 3](#_Toc415735968)

[1.1 Purpose 3](#_Toc415735969)

[1.2 Intended Audience 3](#_Toc415735970)

[1.3 Document Scope 3](#_Toc415735971)

[1.4 Functional Rules 3](#_Toc415735972)

[1.5 Assumptions/Restrictions 3](#_Toc415735973)

[1.6 Documents Structure 4](#_Toc415735974)

[1.7 Additional Resources 4](#_Toc415735975)

[2. High Level Design Diagram 6](#_Toc415735976)

[2.1 Overview 6](#_Toc415735977)

[2.2 Sequence Diagram 7](#_Toc415735978)

[2.3 Use Case Diagram 7](#_Toc415735979)

[3. Detailed Application Design 9](#_Toc415735980)

[3.1 Class Diagram 9](#_Toc415735981)

[3.2 Beans 9](#_Toc415735982)

[3.3 Facade 9](#_Toc415735983)

[3.4 Strategy 9](#_Toc415735984)

[3.5 Controller 9](#_Toc415735985)

[3.6 View/JSP 10](#_Toc415735986)

[4. Database architecture 11](#_Toc415735987)

[4.1 ER Diagram 11](#_Toc415735988)

[5. Exception Handling and Logging 12](#_Toc415735989)

[5.1 Exception Scenarios 12](#_Toc415735990)

[5.2 Exception Handling 12](#_Toc415735991)

[5.3 Logging 12](#_Toc415735992)

[6. Installation and Configuration 13](#_Toc415735993)

[6.1 Installation steps for the addon 13](#_Toc415735994)

[6.2 Configuration steps for the addon 13](#_Toc415735995)

[7. Appendix A *– Glossary of terms* 14](#_Toc415735996)

1. Introduction

Purpose

This document is the Technical Design Document which provides information of hotfolder addon and steps to configure the module. It covers:

1. High Level Design
2. Application Architecture
3. Database Architecture
4. Installation Steps

Intended Audience

This document is aimed at Hybris developers and Technical members who want to implement the hotfolder addon in Hybris.

Document Scope

This document covers the details about the sprint 1 hot folder addon.

Functional Rules

* The hot folder permits to import some data about stocklevels with file format CSV
* All fields of different types can be considered

Assumptions/Restrictions

This module has been created for version 5.4.0.0.

Display: The hot folder module provide the directory where the CSV files have to be put to automatically import data in the system

Import: The different template we provide in CSV files filled

Export: No export is available

Documents Structure

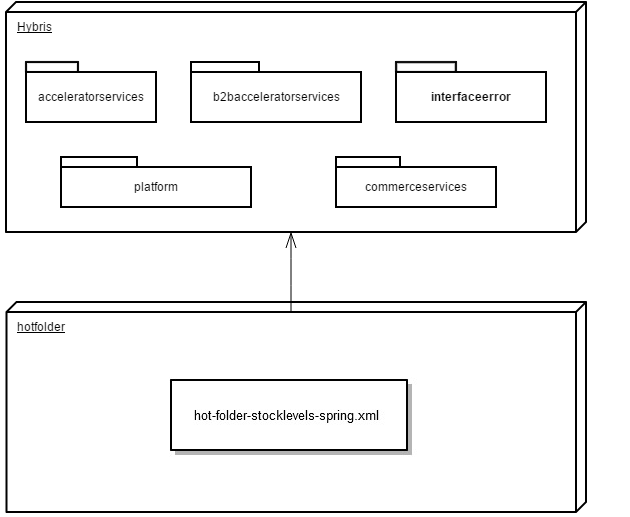
|  |  |  |
| --- | --- | --- |
| Section | | Description |
| 1 | Introduction | Describes the purpose and intended audience of the document and explains its structure. |
| 2 | High Level Design | Overview of the module. |
| 3 | Detailed Application Design | Application architecture applicable to this module. |
| 4 | Detailed Database Design | Conceptual and logical architecture applicable to this module. |
| 5 | Exception Management & Logging | Information about the Exception handling and logging. |
| 6 | Installation & Configuration | Steps to install and configure the module. |
| 7 | Appendices | Supporting information. |

Additional Resources

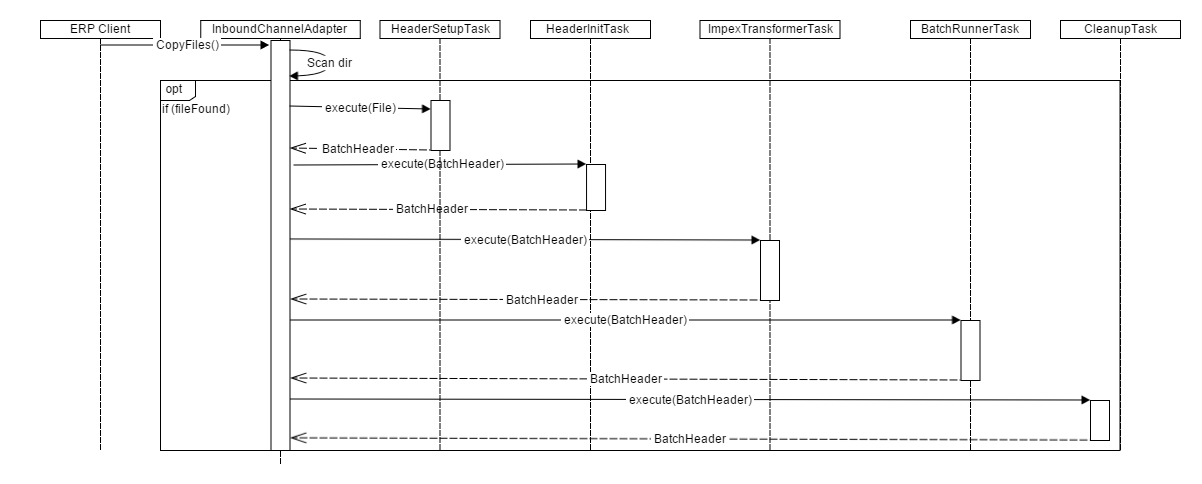
| # | Description | URL | Purpose |
| --- | --- | --- | --- |
| 1 | Wiki Portal | <https://wiki.hybris.com> | To gather more information |
| 2 | Coconet | <https://coconet2.capgemini.com/sf/go/doc3313610?nav=1> | To refer functional specifications |
| 3 | Data importation | <https://wiki.hybris.com/display/accdoc/Data+Importing+Capability+in+the+Commerce+Accelerator> | Documentation of import process |
| 4 | File stockLevelsData with header |  | Template of header stockLevelsData |
| 5 | File stockLevelsData with data |  | Sample of stockLevelsData with data |

1. High Level Design Diagram

Overview



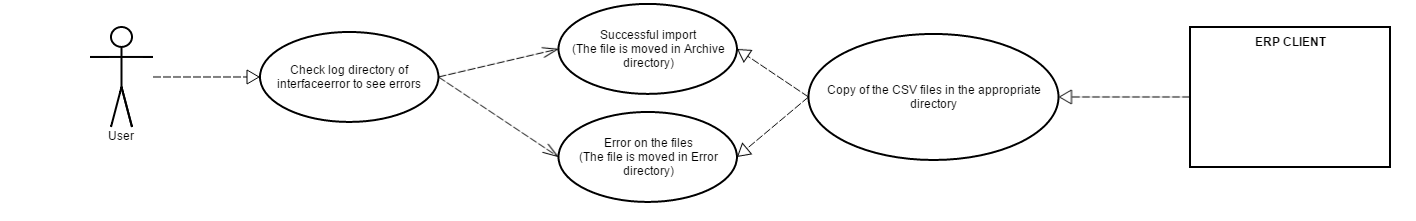
Sequence Diagram

Sequence of events during the hot folder process

|  |  |  |
| --- | --- | --- |
| **#** | **Component** | **Description** |
| 1 | **ERP client** | The system of the client which provide the CSV files |
| 2 | **InboundChannelAdapter** | Spring integration scans folder and looks for new files that matches filename prefix |
| 3 | **HeaderSetupTask** | The task which create BatchHeader initialize import process with parameters Catalog and Net |
| 4 | **HeaderInitTask** | The task which initialize import process with parameters Lang and SequenceID |
| 5 | **ImpexTransformerTask** | The task that is responsible of the conversion into IMPEX format |
| 6 | **BatchRunnerTask** | The task which does the import process |
| 7 | **CleanupTask** | The task which clean the directory “process” and move files in directory “error” or “archive” |

Use Case Diagram

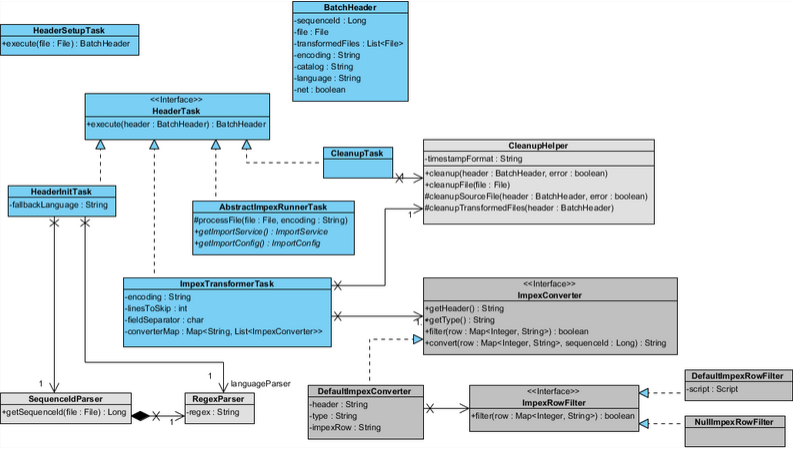
Use case diagram for the hot folder stocklevels addon



**Description:** User checks the result of the import process starts from ERP CLIENT.

1. Detailed Application Design

Class Diagram



Beans

**HeaderSetupTask:** The task which initialize import process with parameters.

**HeaderInitTask:** Define the langage of the import file and verify that the file isn’t a duplicate

**ImpexTransformerTask:** Define some variables for the conversion and do this.

**CleanupTask:** Clean the import files and move them into error folder or archive error.

Facade

N.A

Strategy

N.A

Controller

N.A

View/JSP

N.A

1. Database architecture

ER Diagram

**When you find example with (code), you have to import the object relation before the stocklevels data.  
  
Stocklevels ER Diagram:**

| **Name** | **Description** | **Type** | **Default Value** | **Mandatory** | **Example** |
| --- | --- | --- | --- | --- | --- |
| productCode | Product ID | String | / | TRUE | 107701 |
| available | Number of product available | java.lang.Integer | / | TRUE | 8 |
| inStockStatus | Status of the stock | InStockStatus | / | FALSE | notSpecified |
| maxPreOrder | Maximum preorderable items for the product | java.lang.Double | 1 | TRUE | 1 |
| nextDeliveryTime | Next delivery time | java.util.Date | / | FALSE | 03/06/2015 |
| overSelling | Number of product which can be over selled | java.lang.Integer | 0 | TRUE | 5 |
| preOrder | Preorderable items for the product | java.lang.Integer | / | TRUE | 0 |
| releaseDate | Date of release | java.util.Date | / | FALSE | 03/06/2015 |
| reserved | Number of product reserved | java.lang.Integer | 0 | TRUE | 0 |
| warehouse | Name of the warehouse | Warehouse | default | TRUE | pw\_warehouse\_s |
| treatNegativeAsZero | Value automatically set to 0 if negative | java.lang.Boolean | / | TRUE | true |

1. Exception Handling and Logging

Exception Scenarios

The module interfaceerror log all actions that can handle errors. It is log using Log.error() method in the log file.

Exception Handling

Generated by de.hybris.platform.impex.jalo.imp.DefaultImportProcessor

|  |  |
| --- | --- |
| **Exception** | **Message** |
| UnresolvedValueException | Exception caught when :   * error finding existing item * no existing item found for update * cannot create due to unresolved mandatory/initial columns |
| InsufficientDataException | Exception caught when :   * target type is not permitted by current header – {header} * cannot import localized column without default session language |
| ImpexException | Exception caught when :   * unknow mode (INSERT | UPDATE | INSERT\_UPDATE | REMOVE) * batch inserts are not supported – found more than one item for value line {valueLine} * Configured type {type} is abstract and you want to insert {valueLine}. Please specify the subtype of the item you want to create at the first column of the value line or change the configured type * Composed type {code} of found item with PK {PK} is not an instance of specified type {typecode} * Composed type {code} of found item with PK {PK} is not permitted in Insert\_Update mode, because of {typeErrorMsg} |
| HeaderValidationException | Exception caught when :   * cacheUnique modifier is not allowed for Singleton types, please remove cacheUnique modifier |

Logging

* For logging in the advanced stocklevels interface search module Log4j will be used.
* The logging level will be set at INFO level.
* The exceptions are logged using Log.error() method.

1. Installation and Configuration

Installation steps for the addon

Add the hotfolder module dependency in hybris/Assets/hybris/config/localextension.xml file as follows.

<extension name="hotfolder"/>

Next, Copy **both hotfolder and interfaceerror** module directory into hybris/bin/custom.

To finish, execute the following commands marked in green using command prompt.

Cd hybris.directory/bin/platform

# **Note:** hybris.directory is the installation location of the hybris

setantenv.bat

ant all

Configuration steps for the addon

If you want to modify the parameters of the hotfolder module, you can find these in the **project.properties** file of the module.

You can change the Frequency scan of the poller directory, the default name of the catalog and the default version of the catalog.

# Frequency in ms of the directory scan

hotfolder.stocklevels.poller.rate=1000

# The name of the default catalog

hotfolder.stocklevels.default.catalog=powertoolsProductCatalog

# The version of the product default catalog

hotfolder.stocklevels.default.catalogVersion=Staged

7. Appendix A *– Glossary of terms*

|  |  |
| --- | --- |
| **Term/Abbreviation/Acronym** | **Meaning** |
| B2B | Business to Business |