

Hybris commerce suite

Interface error module

Author: Benoit Durant de la Pastellière

Date: 21-03-2014

Sommaire

[1. Overview 2](#_Toc383162917)

[1.1 Problematic 2](#_Toc383162918)

[1.2 Solution 2](#_Toc383162919)

[2. Utilization 3](#_Toc383162920)

[2.1 Installation 3](#_Toc383162921)

[2.2 Utilization 3](#_Toc383162922)

[2.2.1 Log4j configuration 3](#_Toc383162923)

[2.2.2 Hot-folder configuration 3](#_Toc383162924)

[3. Error list messages : 4](#_Toc383162925)

[(generated by de.hybris.platform.impex.jalo.imp. DefaultImportProcessor ) 4](#_Toc383162926)

1. Overview

Problematic

During imports, issues that compromises correct import of data into the hybris system is lost in a console log cloud. Rejected files are backed up in an error folder, which is not only hard to find but doesn’t provide a relevant error message nor a fix clue making an import fail nearly impossible to trace and fix.

Solution

The impex module takes place in the context enabling tracing, managing and clarifying these causes of error.

These errors are shown using two folders “reject” and “log” containing csv files with the following information:

Reject : these files show the line in the source file

Log : these files show a way to find out failing lines in the source file, and the cause of the fail in the following format :

* original file: {original\_file\_name};line:{ original\_file\_line};reject file: {absolute\_reject\_file\_path};line {reject\_file\_line};message: {error\_cause}

These files get the same extension as the original one and the same file name suffixed by their nature (\_reject or \_log).

They’re also a single file mode output following this format :

* original file: {original\_file\_name};line: {original\_file\_line};message: {error\_cause};original line:{original\_line}
* This single file mode can be useful to achieve mail log

1. Utilization

Installation

* Register the plugin in **config/localextensions.xml**

<extension name="**interfaceerror**" />

Configuration

Log4j

* Override this default output configuration in **config/local.properties**

log4j.logger.com.generic.interfaceerror.logger=debug, **INTERFACE\_ERROR\_REJECT, INTERFACE\_ERROR\_LOG, INTERFACE\_ERROR\_REJECTLOG**, CONSOLE

//!\\ INTERFACE\_ERROR\_LOG need the reject filename requiring **INTERFACE\_ERROR\_REJECT to be called BEFORE INTERFACE\_ERROR\_LOG!**

* Override this default configuration for output directory/file in **config/local.properties**

log4j.appender. INTERFACE\_ERROR\_REJECT.**directory** =${INTERFACE\_ERROR\_LOG\_DIR}/reject

log4j.appender. INTERFACE\_ERROR\_LOG.**directory** =${INTERFACE\_ERROR\_LOG\_DIR}/log

log4j.appender. INTERFACE\_ERROR\_REJECTLOG.**file** =${INTERFACE\_ERROR\_LOG\_DIR}/log.csv

* **For INTERFACE\_ERROR\_REJECT and INTERFACE\_ERROR\_LOG, only the folder matters**, filename is resolved at runtime.

Hot-folder configuration

Create or edit you existing {module}/resources/{module}/integration/**hot-folder-\*.xml** files :

* Change the output channel of the service-activator of the HeaderSetupTask to **interfaceerrorBatchFilesHeaderInit**. i.e. :

<!-- 3) transform csv to impex file -->

<int:service-activator input-channel="batchFilesBiomerieuxProc" **output-channel="interfaceerrorBatchFilesHeaderInit"**

ref="biomerieuxHeaderSetupTask" method="execute" />

* In you Impex Converter bean declaration (beans with header and impexRow properties)
  + Add as parent bean **interfaceerrorImpexConverter** . I.e. :

<bean id="batchAddressConverter" **parent="interfaceerrorImpexConverter"** >

<property name="header">

<value> … </value>

</property>

<property name="impexRow">

<value> … </value>

</property>

</bean>

* + In the header definition, add the following custom processor **interfaceerrorImportProcessorClassName** :

$proc=processor=#{**interfaceerrorImportProcessorClassName**}

INSERT\_UPDATE {Objet}[**$proc**]

**You’re done !**

1. Error list messages :

**(generated by de.hybris.platform.impex.jalo.imp.DefaultImportProcessor )**

UnresolvedValueException (entity not found) :

* error finding existing item
* no existing item found for update
* cannot create due to unresolved mandatory/initial columns
* error creating item despite missing mandatory/initial columns

InsufficientDataException :

* target type is not permitted by current header – {header}
* cannot import localized column without default session language

ImpExException

* unknown 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

* cacheUnique modifier is not allowed for Singleton types, please remove cacheUnique modifier

1. Migrating from bmx to generic version (before 4/11/2014)

config\localextensions.xml

<extension name="bmximpex" />

Has becomed :

<extension name="interfaceerror" />

hot-folder-\*.xml files

{module}/resources/{module}/integration/hot-folder-\*.xml files

output-channel="bmxBatchFilesHeaderInit"

**has becomed :**

output-channel="interfaceerrorBatchFilesHeaderInit"

$proc=processor=#{bmxImportProcessorClassName}

**has becomed :**

$proc=processor=#{interfaceerrorImportProcessorClassName}

parent="bmxImpexConverter"

**has becomed**

parent="interfaceerrorImpexConverter"

Others (ConcatTool) :

import com.biomerieux.impex.concat.ConcatTool;

**has becomed :**

import com.generic.interfaceerror.concat.ConcatTool;

**as well as in log4j**

log4j.logger.com.biomerieux.impex.concat=debug, CONCATPRICE, CONSOLE

**has becomed**

log4j.logger.com.generic.interfaceerror.concat=debug, CONCATPRICE, CONSOLE

**Capgemini_Slogan_White.pngwww.capgemini.com**

Les informations de cette présentation sont confidentielles et la propriété de Capgemini. ©2012 Capgemini. Tous droits réservés.