hybris software

An SAP Company

hybris software
An SAP Company

hybris Developer Training Part I - Core Platform

Impex

hybris software
An SAP Company

Overview

Syntax and examples

Invoking

ImpEx – Overview

- ImpEx is an out-of-the-box import / export framework
- It's an interface between CSV files and the hybris Commerce Suite's Type System
  - you can "import" instances of types from CSV.
  - you can "export" instances of types into CSV.
- You can create, update, export, and remove items

ImpEx – Typical fields of use

- In live operation:
  - to import customer data into a production system
  - to synchronize data with other systems, such as an ERP or LDAP
  - to create backups
  - to update data at runtime
  - can be run from CronJobs
- In migrations:
  - to migrate data from one hybris installation to another
- In development:
  - to import sample data (e.g. on system initialization)
  - to import test data into testing system

ImpEx – Features

- Impex abstracts from database
  - No table information (deployment)
  - No foreign keys (use “business keys,” which we will discuss in a moment)
- Impex simplifies imports
  - The order of the imported data is irrelevant! (Failed lines are retried)
  - Validation constraints may be disabled

```
impex.legacy.mode=true
```

- ImpEx concerns
  - no transactional imports
  - Performance – use multithreaded imports:

```
impex.import.workers=4
```

- Note: ImpEx does not provide XML import out-of-the-box

hybris software
An SAP Company

Overview

Syntax and Examples

Invoking

Syntax Basics

Header syntax:

```
Operation      itemType;  attributes(refAttr)[modifiers];...

INSERT         Product;   code; name[lang=en];
UPDATE         Product;   code[unique=true]; name[lang=en];
INSERT_UPDATE  Customer;  customerID[unique=true]; groups(uid);
REMOVE         Media;     code[unique=true];
```

Data row syntax:

```
;attr1value; attr2value; ...
;iphone5; Apple iphone 5;
;Drew; customergroup;
;iphone5Pic;
```

Basic syntax example

```
INSERT_UPDATE Promo; code[unique=true]; name[lang=en]; country(code)
;BKCampaign1;Burger King Antarctica Launch; AQ
;iphone5China;Apple iphone 5 China Campaign; CN
```

- Key points:
  - The `code[unique=true]` is so called “key attribute” or “business key”. Impex will search for product with code “`BKCampaign1`” before triggering import. In this example we expect to find 1 or none.
  - The `name[lang=en]`indicates language of provided value. Only valid for localized attributes.
  - The `country(code)` is a reference to another item using its code (“business key”) In this example, the *country* property of Promo item “`BKCampaign1`” is linked to another hybris item whose code is “`AQ`”

ImpEx Syntax Elements

- Macros
  - Allows aliases to stand in for frequently used statements
- BeanShell
  - Allows script to be added to a CSV file.
  - Predefined hooks `beforeEach`, `afterEach`, `getLastImportedItem()` etc.
- Translators
  - Implement custom ImpEx logic e.g to import *medias* (binary files).
- Inclusion of data
  - Allows you to split your ImpEx operations over several files.
- Collections and HashMaps:
  - Allows you to use these types as attributes
- Different validation modes for export
  - E.g the mode “Strict (Re)Import” ensures that the export is re-importable

Catalog example

```
$catalogVersion=catalogVersion(Catalog(id),version)[unique=true]

INSERT_UPDATE Product; code[unique=true]; name[lang=en];
unit(code); $catalogVersion

;W580i;Sony Ericsson W580i; pieces; Default:Staged

;iphone5;Apple iphone 5; pieces; Default:Online
```

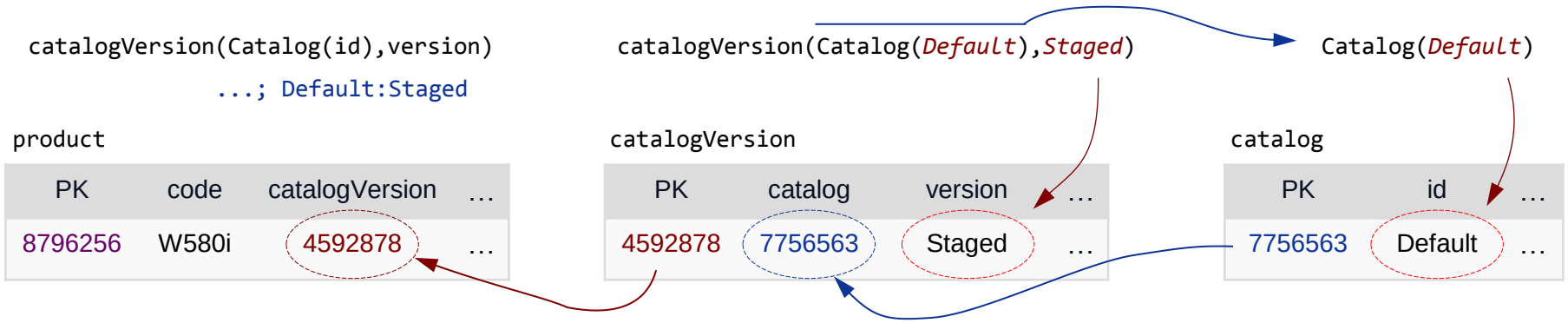
- Key Points:
  - This example uses a macro, which is substituted in the header verbatim.
  - The Product is considered to have a composite key, since two fields in the header are listed as unique (code and catalogVersion).
  - For catalog-aware items (e.g. product), we must specify the catalog version for the item. Since this is a double-key reference, we separate the fields with a colon (:).

Catalog reference details

```
$catalogVersion=catalogVersion(Catalog(id),version)[unique=true]
INSERT_UPDATE Product; code[unique=true]; name[lang=en]; unit(code); $catalogVersion
;W580i;Sony Ericsson W580i; pieces; Default:Staged
;iphone5;Apple iphone 5; pieces; Default:Online
```

References

- The product item references a catalogVersion item, which is identified using two keys: a catalog reference and a *version* string. The catalog reference, in turn, is identified by an *id* string.



Using macros and defaults

```
$prodCat=myCatalog
$version=Staged
INSERT Category;code;catalogVersion(catalog(id),version)
;cars;$prodCat:$version
;convertibles;$prodCat:$version
```

```
$catVersion=catalogVersion(catalog(id[default=$prodCat]),version[default=$v
ersion])
INSERT Category;code;$catVersion
;cars;
;convertibles;
```

- Notes
  - macros can be used in both header and data rows
  - use default values to simplify data rows

More examples

- Use ‘append’ mode to avoid overriding existing references

```
INSERT_UPDATE Employee; uid[unique=true]; groups(uid)[mode=append]
;Drew; approvers,dummygroup,reviewers
```

- Use ‘translators’ for custom interpretation of imported values

```
INSERT_UPDATE Employee;@password[translator=PasswordTranslator]
;aVeryStrongPassword;
```

```
INSERT_UPDATE Media; @media[translator=MediaDataTranslator]
;/path/to/my/picture.jpg;
```

Advanced example

Batch update

```
UPDATE Product [batchmode=true]; itemType(code)[unique=true];status
;Product; approved
```

Selective export

```
"#% impex.setTargetFile( ""Product.csv"" );"
INSERT_UPDATE Product; code[unique=true]; name[lang=en]
"#% impex.exportItemsFlexibleSearch(
     ""select {pk} from {Product} where {code} like '%happy%'"");"
```

ImpEx Script For Export

- Specify the target file:

```
"#% impex.setTargetFile( ""Product.csv"" );"
```

- Specify the attributes to be exported via an ImpEx header:

```
INSERT_UPDATE Product; code[unique=true]; description[lang=en];
                       name[lang=en]; unit(code)
```

  - You can use the same header for re-import.

- Start the export:

```
"#% impex.exportItems( ""Product"" , false );"
```

- Hint:
  - You may use the “Script Generator” in the hMC.

hybris software
An SAP Company

Overview

Syntax and examples

Invoking

Where Can You Launch an Import?

- In the hybris Administration Console
  - Test area for ImpEx scripts
  - Multiple files cannot be imported by a single ImpEx script
  - No external data is allowable
  - Limitted configuration possibilities
- In the hybris Management Console (hMC)
  - Create an **`ImpExImportCronJob`**
- Via the API
  - You can use the **`ImportService`**

Where Can You Launch an Export?

- In the hybris Administration Console
  - Test area for ImpEx scripts
- In the hybris Management Console
  - Select search results and export them via the context menu
  - Create an **`ImpExExportCronJob`**.
- Via the API
  - Use the **`ExportService`**
  - Create an **`ImpExExportCronJob`**

Quiz Questions

1. What data can you import with ImpEx?

2. How do you trigger imports?

3. What happens if your header lines are broken?

4. What happens if some data lines are broken, e.g. referencing unknown data?