**Usecase 1: Update Revision name when classification object is created**

**BMIDE :**

Extension Name : CT4\_ClassificationforModul

Extension Type: Operation

BO: IMAN\_Classification

Operation: GRM\_Create

Action : Post Action

Graphical user interface, application

Description automatically generated

**DLL: libCT4Customlib.dll**

Function Name: int CT4\_ClassificationforModul(METHOD\_message\_t\* msg, va\_list args)

1. This Extension is triggered during Classification object creation
2. Extension checks CT4\_CLASSIFICATION\_CLASS preference for valid Item revision
3. Get user data parameter from the storage class
4. Based on the user parameter get the key Lov attributes to be extracted.
5. Extract the key LOV values from the classification object.
6. Update item revision name with key lov values

Format : <Item Rev Name>~<attribute1>~<attrbiute2>……….

**Usecase 2: Update item revision name on modification of Userdata parameter of storage class**

**ITK exe: UpdateClassificationInfo.exe**

1. This exe will check all classification storage class modified after last execution.
2. It finds all the revision under the classification storage class
3. Exe checks CT4\_CLASSIFICATION\_CLASS preference for valid Item revision
4. Get user data parameter from the storage class
5. Based on the user parameter get the key Lov attributes to be extracted.
6. Extract the key LOV values from the classification object.
7. Update item revision name with key lov values

Format : <Item Rev Name>~<attribute1>~<attrbiute2>……….

**Execution:**

1. Open TC Command promptGraphical user interface, application

   Description automatically generated
2. Execute updateclassification\_first.bat in TC command promote

Text

Description automatically generated

1. Verify below text file is created.

Graphical user interface, text, application, email

Description automatically generated

After Step 4 : Modify the storage class userdata parameter

1. Execute updateclassification.bat to update Item revision name.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

1. Check logfile in Log folder for IR’s which got updated.

Graphical user interface

Description automatically generated with medium confidence

**Custom Date change and execution**

Step 1: Open below file and change the date highlighted. Maintain the format

Graphical user interface, text, application

Description automatically generated

Step 2: Execute updateclassification.bat to update Item revision name.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

**Usecase 3: Update item revision name when classification object is updated**

**Custom JAR:**

Jar name: ct4.svnamebyclass.rac\_13000.1\_V01

Copy to %TC\_BIN%\portal\plugins

Java Class: G4MSaveParsaPostHook

1. Created a custom hook “g4mSave.customPostHook.COMMAND” which extends Save operation on classification object save in RAC
2. Overrided executeCommand for GM4SAVE. This method is tiggered on save operation in classification module.
3. This method checks CT4\_CLASSIFICATION\_CLASS preference for valid Item revision
4. Get user data parameter from the storage class
5. Based on the user parameter get the key Lov attributes to be extracted.
6. Extract the key LOV values from the classification object.
7. Update item revision name with key lov values

Format : <Item Rev Name>~<attribute1>~<attrbiute2>……….

**Usecase 4: Clear Revision name when classification object is deleted.**

**BMIDE :**

Extension Name : CT4ClassificationObjDel\_PreCond

Extension Type: Operation

BO: IMAN\_Classification

Operation: IMAN\_delete

Action : Pre Condition

Graphical user interface, application

Description automatically generated

**DLL: libCT4Customlib.dll**

Function Name: int CT4ClassificationObjDel\_PreCond(METHOD\_message\_t\* msg, va\_list args)

1. This Extension is triggered before deleting the Classification object
2. Extension checks CT4\_CLASSIFICATION\_CLASS preference for valid Item revision
3. Collect the item revision and classification object which are being deleted.
4. Collected data is pushed to map. This map is used in post action for clearing the name on item revision.

**BMIDE :**

Extension Name : CT4ClassificationObjDel\_PostAct

Extension Type: Operation

BO: IMAN\_Classification

Operation: IMAN\_delete

Action : Post Action

Graphical user interface, text, application

Description automatically generated

**DLL: libCT4Customlib.dll**

Function Name: CT4ClassificationObjDel\_PostAct(METHOD\_message\_t\* msg, va\_list args)

1. This Extension is triggered after deleting the Classification object
2. Extension checks CT4\_CLASSIFICATION\_CLASS preference for valid Item revision
3. Use IR collected in the pre-condition extension.
4. Clear the classification details from item revision name.