***Rental Portal***

**Documentation**

**on**

**Scheduled Activities**

**Version**

| **Version No.** | **Created / Updated By** | **Date** | **Changes** |
| --- | --- | --- | --- |
| 0.1 | Abhishek Kumar Shaw | 02 Nov 2022 | Initial document |

**Purpose**

This Document is meant to describe the workings of the scheduled tasks that are run for the rental portal.

**Scope**

This Document covers the processes of all the scheduled tasks in the Rental portal. It also describes the relationship of the Database with the schedular processes.

**InsertContractmaster** Class Name - Scheduler

Function Name - contractmaster

XML File called - *GCPL\_X\_RUEContMast.XML*

TABLE Name- ***ruep\_contract\_master***

Process :-

* Loading of XML file by creating new object of SimpleXMLElement library
* extracting data in array format and storing it in array variable
* Checking if array is empty or not
* If it isn’t empty then checking contract\_no field emptiness
* If not empty then putting value of contract\_no

If empty then replacing it by ‘ ‘

* Then looping over array with checking all fields emptiness and storing key -> value pair in a data variable
* Inserting the data variable in scheduler model->insert\_contract\_master
* Deleting of xml file

| **Fields in XML** | **Fields in DB** |
| --- | --- |
| contract\_no | cont\_no |
| cust\_po | cust\_po |
| cust\_po\_date | cust\_po\_date |
| sold\_party | sold\_party |
| pay\_term | pay\_term |
| emp\_resp | emp\_resp |
| contract\_date | cont\_date |
| hire\_date | hire\_date |
| de\_hire\_date | de\_hire\_date |
| agreement\_tenure | agreement\_tenure |
| contract\_sign | cont\_sign |
| contract\_status | cont\_status |
| werks | werks |
| bzirk | bzirk |
| srno | srno |
| bus\_div | bus\_div |
| ind\_div | ind\_div |
| busgrp\_code | busgrp\_code |
| matnr | matnr |

| rental\_rate | rental\_rate |
| --- | --- |
| usage\_charge | usage\_charge |
| usage\_disc | usage\_disc |
| netwr | netwr |
| cgst | cgst |
| sgst | sgst |
| igst | igst |
| bts\_check | bts\_check |
| pdc\_check | pdc\_check |
| seq\_depo\_check | seq\_depo\_check |
| helper\_check | helper\_check |
| trans\_check | trans\_check |

**InsertAssetmaster**

Class Name - SchedulerFunction Name - assetmaster

XML File called - *GCPL\_X\_RUEAsstVal.XML*

TABLE Name - ***ruep\_asset\_master***

Process :-

* Loading of XML file by creating new object of SimpleXMLElement library
* extracting data in array format and storing it in array variable
* Checking if array is empty or not
* If it isn’t empty then checking contract\_no field emptiness
* If not empty then putting value of ANLN1

If empty then replacing it by ‘ ‘

* Then looping over array with checking all fields emptiness and storing key -> value pair in a data variable
* Inserting the data variable in scheduler model->insert\_asset\_master
* Deleting of xml file

| **Fields in XML** | **Fields in DB** |
| --- | --- |
| ANLN1 | asset\_code |
| ANLKL | asset\_class |
| ZUPER | acquisitio\_period |
| ZUJHR | acquisition\_year |
| ZUGDT | first\_acquisition\_date |
| DEAKT | asset\_deactivation\_date |
| AKTIV | capitalised\_date |
| LIFNR | vendor\_code |
| LAND1 | country\_of\_origin |
| HERST | manufacturer |
| CAUFN | io\_no |
| WERKS | plant |
| STORT | location |
| KANSW | asset\_cost |
| KNAFA | accumulated\_depn |
| NAFAG | depn\_for\_current |
| WDV | wdv\_value |
| INVNR | id\_no |

**InsertCustmaster**

Class Name - Scheduler

Function Name - custmaster

XML File called - *GCPL\_X\_RUECustMast.XML*

TABLE Name - **ruep\_cust\_master**

Process :-

* Loading of XML file by creating new object of SimpleXMLElement library
* extracting data in array format and storing it in array variable
* Checking if array is empty or not
* If it isn’t empty then checking contract\_no field emptiness
* If not empty then putting value of KUNNR (cust\_code)

If empty then replacing it by ‘ ‘

* Then looping over array with checking all fields emptiness and storing key -> value pair in a data variable
* Inserting the data variable in scheduler model->insert\_cust\_master
* Deleting of xml file

| **Fields in XML** | **Fields in DB** |
| --- | --- |
| NAME1 | cust\_name |
| KUNNR | cust\_number |
| REGIO | cust\_region |
| STRAS | cust\_address |
| TEL\_NUMBER | cust\_contact\_no |
| KUKLA | cust\_classification |
| BRSCH | cust\_key |
| BRAN1 | cust\_code |
| GFORM | cust\_legal\_status |
| STCD3 | cust\_gstin\_no |
| PANNO | cust\_pan\_no |

**InsertEquipmaster**

Class Name - Scheduler

Function Name - equipmaster

XML File called - *GCPL\_X\_RUEEquiMast.XML*

TABLE Name - **ruep\_equipment\_master**

Process :-

* Loading of XML file by creating new object of SimpleXMLElement library
* extracting data in array format and storing it in array variable
* Checking if EQUNR / (equip\_sl\_no) is set or not in array  
  If set then create a newArray of data in that XML.
* Checking newArray emptiness  
  If not empty then looping over array with checking all fields emptiness and storing key -> value pair in a data variable
* Inserting the data variable in scheduler model->insert\_cust\_master
* Deleting of xml file.

| **Fields in XML** | **Fields in DB** |
| --- | --- |
| EQUNR | equip\_sl\_no |
| ANLN1 | equip\_asset\_code |
| ZMODEL | equip\_model |
| EQKTU | equip\_sr\_no\_desc |
| INVNR | equip\_id\_no |
| BAUJJ | equip\_mfg\_year |
| MATNR | equip\_material |
| EQTYP | equip\_category |
| ZMAKE | equip\_manufacturer |
| ZINDV | equip\_industry\_division |
| ZSBU | equip\_sbu |
| ZBUS\_DIV | equip\_business\_division |
| ZPRODUC | equip\_product\_type |
| ZPRODST | equip\_product\_status |
| ZSALIND | equip\_sale\_indicator |
| ZTERRI | equip\_territory\_indicator |
| BZIRK | equip\_territory |
| ZANNUAL | equip\_latest\_smu |
| ZLOCDS | equip\_location |

| WERKS | equip\_planning\_plant |
| --- | --- |
| ZZCHASNO | equip\_chassis\_no |
| ZENGSR | equip\_engine\_sr\_no |
| ZZFITVAL | equip\_fitness\_validity\_date |
| ZZREGNO | equip\_registration\_no |
| ZZINSNO | equip\_insurance\_no |
| ZZINSVALDAT | equip\_insurance\_validity\_date |
| ZZREGLOC | equip\_registered\_location |

**InsertInductionmaster**

Class Name - Scheduler  
  
 Function Name - inductionmaster

XML File called - *GCPL\_X\_RUEInducMast.XML*

TABLE Name - **ruep\_induction\_master**

Process :-

* Loading of XML file by creating new object of SimpleXMLElement library
* extracting data in array format and storing it in array variable
* Checking if ANLN1 / (induc\_asset\_code) is set or not in array  
  If set then create a newArray of data in that XML.
* Checking newArray emptiness  
  If not empty then looping over array with checking all fields emptiness and storing key -> value pair in a data variable
* Inserting the data variable in scheduler model->insert\_induc\_master
* Deleting the xml file.

| **Fields in XML** | **Fields in DB** |
| --- | --- |
| ANLN1 | induc\_asset\_code |
| CHARG | induc\_sl\_no |
| MATNR | induc\_material\_no |
| MAKTX | induc\_material\_desc |
| MENGE | induc\_quantity |
| DMBTR | induc\_amount |
| WAERS | induc\_currency |
| USNAM\_MKPF | induc\_created\_by |
| BUDAT\_MKPF | induc\_date |
| MBLNR | induc\_document\_no |
| MJAHR | induc\_posting\_year |
| BWART | induc\_movement\_type |
| WERKS | induc\_plant |
| LGORT | induc\_storage\_location |

**InsertInvoicemaster**

Class Name - SchedulerFunction Name - invoicemaster

XML File called - .XML

TABLE Name -ruep\_invoice\_master

Process :-

* Loading of XML file by creating new object of SimpleXMLElement library
* extracting data in array format and storing it in array variable
* Checking if array is empty or not
* If it isn’t empty then checking INV\_NO field emptiness
* If not empty then putting value of INV\_NO

If empty then replacing it by ‘ ‘

* Then looping over array with checking all fields emptiness and storing key -> value pair in a data variable
* Inserting the data variable in scheduler model->insert\_invoice\_master
* Deleting of xml file

| **Fields in DB** | **Fields in XML** |
| --- | --- |
| invoice\_no | INV\_NO |
| invoice\_date | INV\_DATE |
| invoice\_line\_item | INV\_LINE\_ITEM |
| document\_type | DOC\_TYPE |
| original\_invoice\_no | ORG\_INV |
| original\_invoice\_date | ORG\_DATE |
| period\_from | PRD\_FRM |
| period\_to | PRD\_TO |
| customer\_po | CUST\_PO |
| customer\_po\_date | CUST\_PO\_DATE |
| contract\_no | CONTRACT\_NO |
| sold\_to\_party | SOLD\_PARTY |
| ship\_to\_party | SHIP\_PARTY |
| terms\_of\_payment | PAY\_TERM |
| employee\_responsible | EMP\_RES |
| plant | WERKS |
| territory | BZIRK |
| region | REGIO |
| machine\_sr\_no | SRNO |

| business\_div\_code | BUS\_DIV |
| --- | --- |
| industry\_div\_code | IND\_DIV |
| business\_group\_code | BUSGRP\_CODE |
| material | MATNR |
| rental\_rate | RENTAL\_RATE |
| extra\_usage\_charge | USAGE\_CHARGE |
| less\_usage\_discount | USAGE\_DISC |
| written\_down\_value | WDV\_COST |
| exw\_cost | EXW\_COST |
| for\_cost | FOR\_COST |
| net\_value | NETWR |
| cgst | CGST |
| sgst | SGST |
| igst | IGST |
| tcs | TCS |
| total\_invoice\_value | TOT\_RENTAL\_REV |

**ZipCreate**

Class Name - SchedulerFunction Name - createZip

Process :-

* Checking if file exists or not  
  If yes then creating a directory in that file path
* Listing the files inside the directory
* Checking the first occurrence of particular fle by it’s name  
  If found then coping the that file to a particular location
* Creating a variable with date and .zip extension
* Creating zip for the particular directory files

**END**

**OF DOCUMENT**