*To enable Activity Tracker for users, “ActivityTrackerAccess” should be 1 in “LoginMaster” table.*

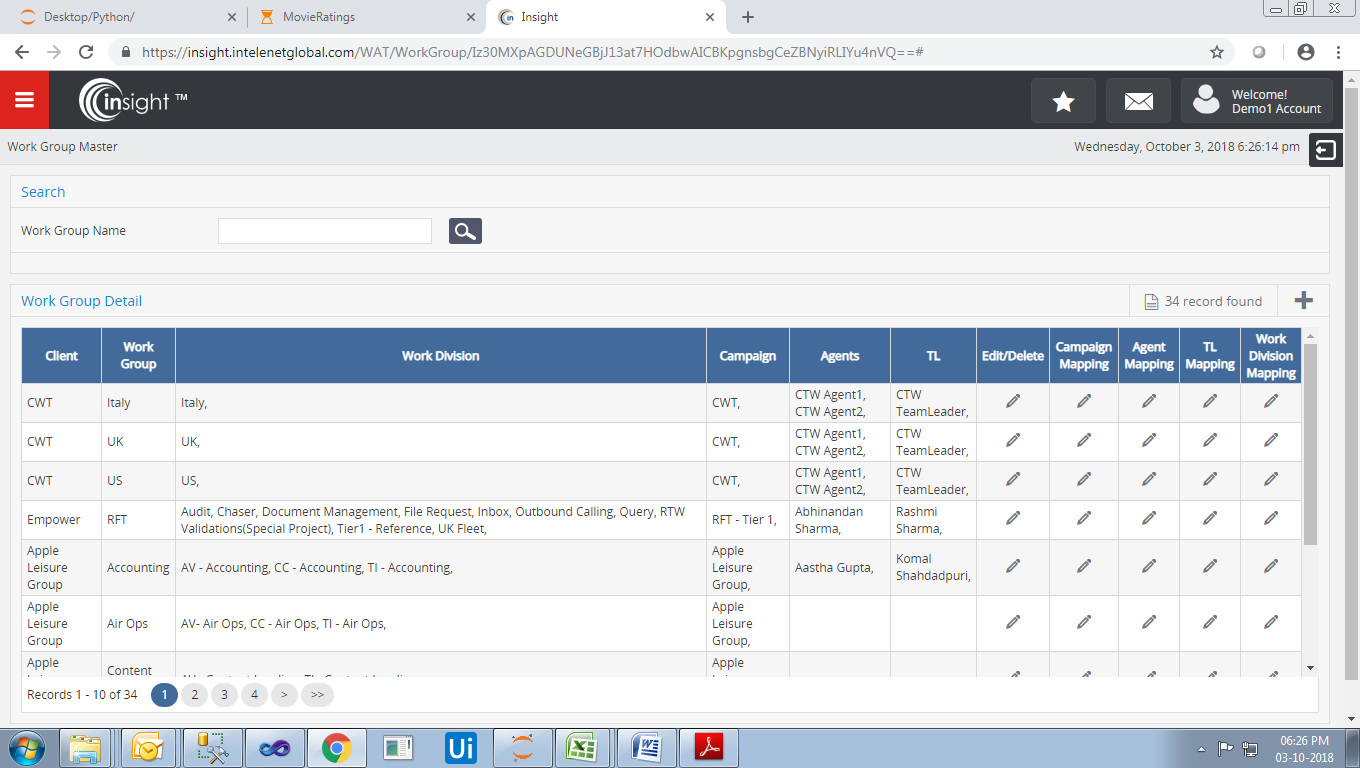
1. **Master Tables of WAT**

1. WAT\_WorkGroupMaster - The module is provided to Configure Work Group for different Clients and Campaign.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkGroupMaster | WorkGMID | Unique Column ID (It will be used as reference in further mapping tables below) |
|  | WorkGroupName | Group Name |
|  | WorkDivision | Will the division be mapped against work group or not(Yes-0/No-1) |

Work Group Master:



**Edit/Delete:** It will edit or delete the work group.

**Campaign Mapping:** Here we can edit the campaign against every work group which is configured. We advised to use single Campaign for a Work Group.

**Agent Mapping:** Agents are mapped against the **Work Group**.

Agent has only access to the work group assigned to him. The mapping of Agent with work group is maintained in “WAT\_WorkGroupAgentDetail” table.

**Note:** We advise to align agent to different but only one single Work Group for a client and multiple campaigns.

**TL Mapping:** TL is mapped against a work group.

TL is mapped to a particular work group. The mapping of TL with work group is maintained in “WAT\_WorkGroupTLDetail” table.

Now let us understand the scenario. Suppose a TL wants only those messages to be visible to the users who are aligned or mapped under that particular TL.

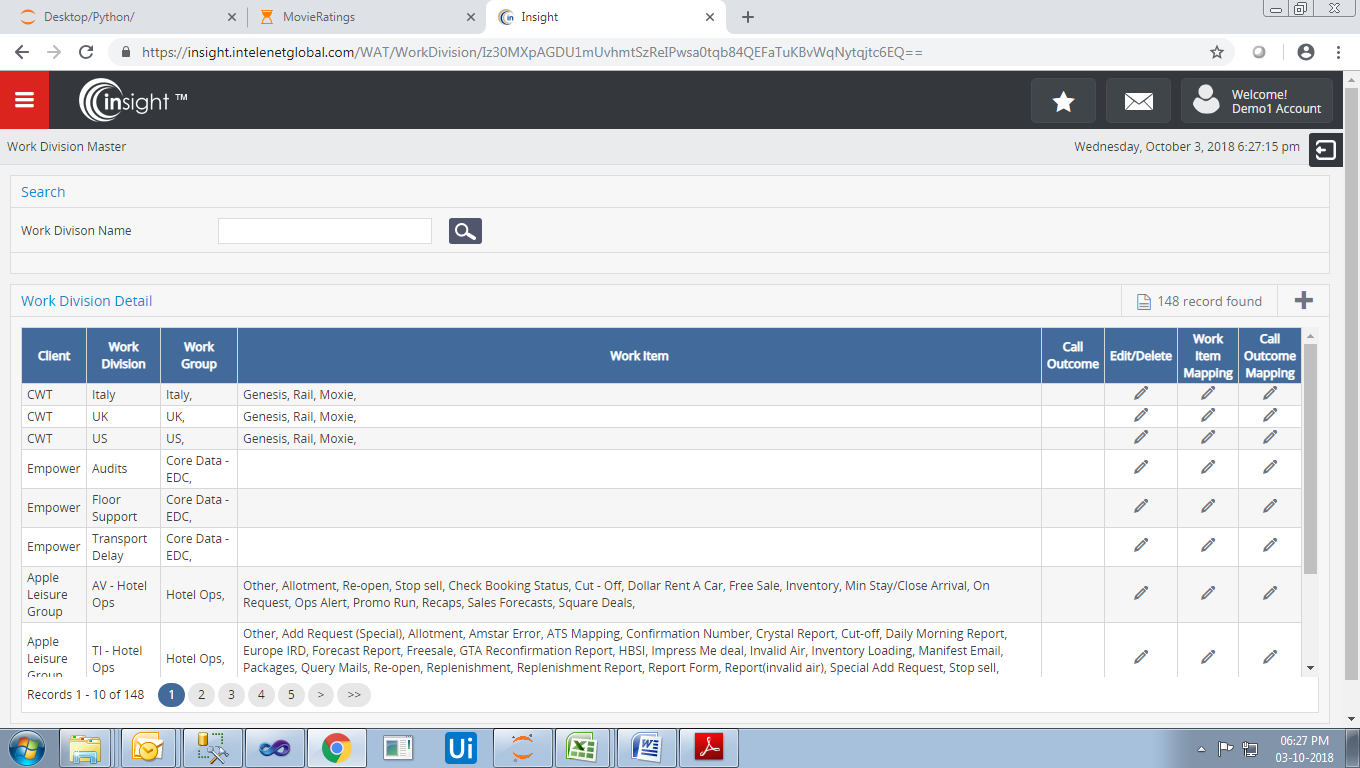
So as to meet this requirement we have added TL mapping.

1. WAT\_WorkDivisionMaster - The module is provided to Add/Update Work Division for different Work Group.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkDivisionMaster | WorkDMID | Unique Column ID (It will be used as reference in further mapping tables below) |
|  | WorkDivisionName | Work Division Name |
|  | WorkItem | Will the item get be mapped against work group or not(Yes-0/No-1)\*\*pooina\*\* |

Work Division Master:



**Edit/Delete:** It will edit or delete the work division.

**Work Item Mapping:** Here we can map work item under the work division.

**Call Outcome Mapping:** The call outcome is mapped against the work division.

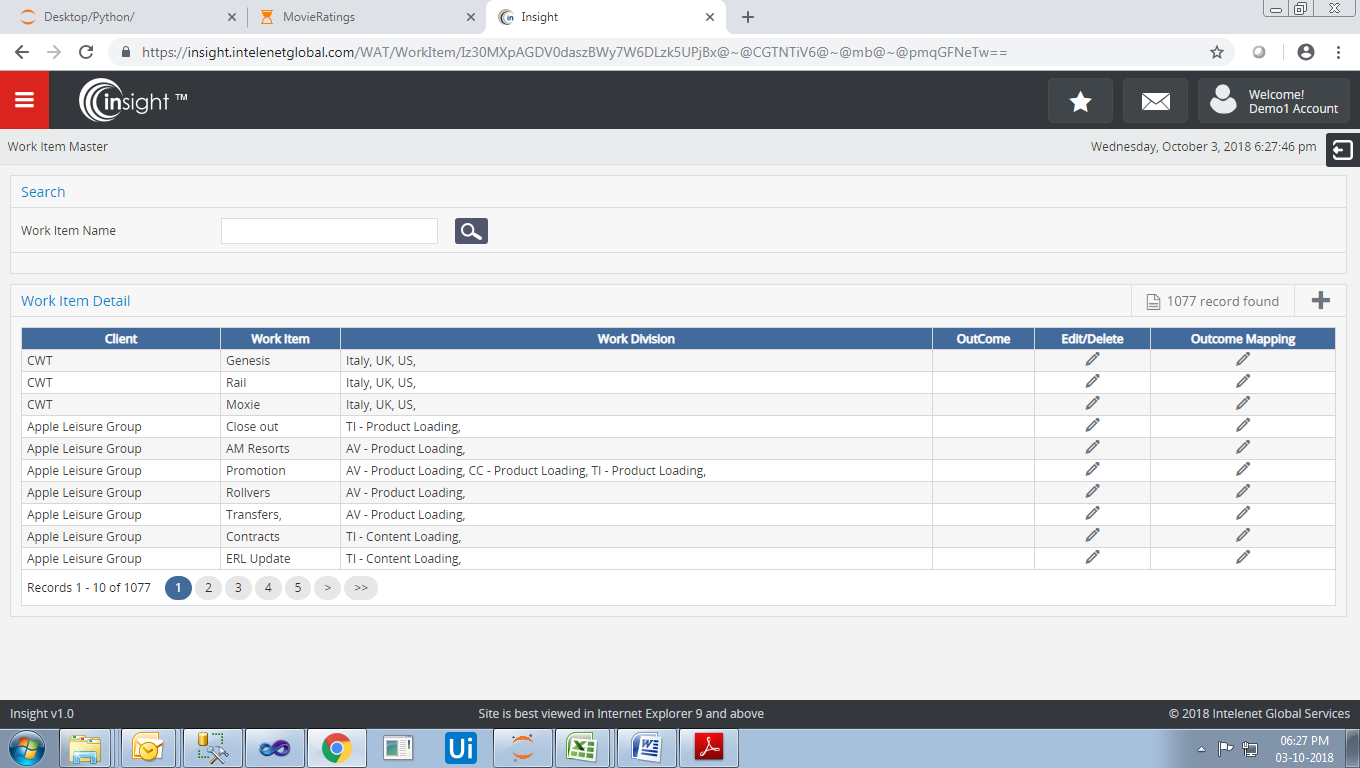
This is captured for calling.

1. WAT\_ WorkItemMaster - The module is provided to Add/Update Work Item for different work group and work division corresponding to campaigns mapped.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkItemMaster | WorkIMID | Unique Column ID (It will be used as reference in further mapping tables below) |
|  | WorkItemName | Work Item Name |
|  | WorkItemIMID |  |
|  | ClientID | ClientId against which work item needs to get mapped |

Work Item Master:



Edit/Delete: It will edit or delete the work item.

Outcome Mapping: Here we map outcome against the work item created.

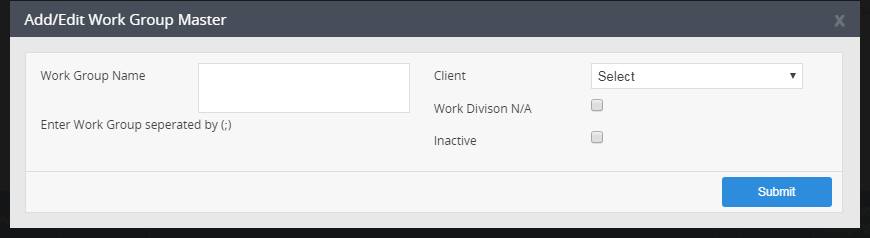
**Note:** The WAT flow moves as Work Group 🡪 Work Division 🡪 Work Item

Further the mapping is done with each other which are explained below in detail.

We can add multiple work group/work division/work item separated by “;”

Please refer to the below screen attached.

Inactive checkbox disables the work group/work division/work item.



1. **Mapping Tables with Master Tables of WAT**
2. WAT\_WorkGroupWorkDivisionDetail - The table maps Work Group ID & Work Division ID.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkGroupWorkDivisionDetail | WorkGWDDID | Unique Column ID |
|  | WorkGMID | Work Group Id |
|  | WorkDMID | Work Division Id |

1. WAT\_WorkGroupCampaignDetail - The table maps for Work Group ID & Campaign ID

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkGroupCampaignDetail | WorkGCDID | Unique Column ID |
|  | WorkGMID | Work Group Id |
|  | CampaignID | Campaign ID for which the work group is mapped |

1. WAT\_WorkGroupAgentDetail - The table maps for Work Group ID & Agent LoginId & GlobalUserId

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkGroupAgentDetail | WorkGADID | Unique Column ID |
|  | WorkGMID | Work Group Id |
|  | LoginMID | Login ID of agent for which the work group is mapped |
|  | GlobalUserID | Global User ID of agent for which the work group is mapped |

1. WAT\_WorkGroupTLDetail - The table maps Work Group Id & TL LoginId & GlobalUserId

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkGroupTLDetail | WorkGTLID | Unique Column ID |
|  | WorkGMID | Work Group Id |
|  | LoginMID | Login ID of TL for which the work group is mapped |
|  | GlobalUserID | Global User ID of TL for which the work group is mapped |

1. WAT\_WorkDivisionWorkItemDetail - The table maps Work Division Id & Work Item Id

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_WorkDivisionWorkItemDetail | WorkDWIDID | Unique Column ID |
|  | WorkDMID | Work Division Id |
|  | WorkIMID | Work item ID for which the work division is mapped |

1. **Master Tables for Disposition in WAT & Configuration Steps:-**

**Note :**

Add one column *WATType* INT NULL in E*meraldHierarchies\_NewInstance.dbo.clients*.

Set WorkTracker Activity for client wise.

*WATType- 1 - Client*

*WATType- 2 - Project*

**1. Redirection for Activity Tracker:**

We have maintained the redirection as per the client against a particular LoginMID.

Table is WAT\_ActivityTrackerScreen.

**Description:**

Table *WAT\_ActivityTrackerScreen* added two columns they are as given below.

**1. ProjectID :** It store projects on the behalf of client. It is required when single client has multiple project but tracker is different . Whenever try to get RedirectURL then is show same tracker which is wrong. Due to this reason we need to Add ProjectID.

**2. Category** : Show RedirectURL on the behalf of Category now not showing client basis.

To insert data using following query.

INSERT INTO dbo.WAT\_ActivityTrackerScreen

( ClientMID ,

ProjectID,

AccessLevel ,

RedirectURL ,

LinkType ,

PageName,

Category

)

VALUES ( 1 ,

203

11 ,

'/Controller/ActionMethod/' ,

1 ,

'Page Name' ,

1

)

**2. AUX Mapping:**

We have maintained the aux mapping as per the client.

Table is WAT\_ActionStatusClientMapping.

Now add one more column in *WAT\_ActionStatusClientMapping*. table, because for fetch the *AuxName* we will go to *WAT\_ActivityTrackerScreen* and fetch *RedirectMID* a/c to client or project or both and get *RedirectMID* and then fetch *AuxName* form *WAT\_ActionStatusClientMapping* on the behalf of *WAT\_ActionStatusClientMapping*

**Note:**

**Insert *RedirectMID* in *WAT\_ActionStatusClientMapping*  on the behalf of ClientId from *WAT\_ActivityTrackerScreen***

To insert multiple data we can refer to following query.

INSERT INTO dbo.WAT\_ActionStatusClientMapping

SELECT ActionSMID ,

ClientMID ,

RedirectMID,

FreezeStatus ,

CreatedDateTime ,

CreatedBy ,

UpdatedDateTime ,

UpdatedBy ,

HostName ,

AuxName ,

ActionStatus ,

ActionGroup ,

StatusMessage ,

SerialId

FROM WAT\_ActionStatusClientMapping

WHERE ActionSCMID IN (1, 2, 3, 4, 5)

3. WAT\_DynamicOutcomeControlMaster - To configure disposition screen against Work Group.

Now let’s suppose that a work group has to be disposed by the user. So this table keeps master entries of the work group created against a specific Client & Campaign.

The Table Name keeps the data for every Client.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_DynamicOutcomeControlMaster | DOCMID | Unique Column ID |
|  | WorkGMID | Work Group Id |
|  | TableName | Table Name in data is saved for a Work Item selected by the user |
|  | ClientID | ClientId against which Work Item is mapped |

4. WAT\_DynamicOutcomeControlMasterDetail – This table maintains the dynamic control details against the master dispositions created in above scenario.

It is used to store the presentable UI corresponding to different Clients and their work groups created.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_DynamicOutcomeControlMasterDetail | DOCDID | Unique Column ID |
|  | DOCMID | Dynamic Control ID (mapped from above table) |
|  | DBColumnName | It should get started from "Col1" which will be found from above TableName column |
|  | LabelName | Label to be displayed on UI |
|  | ControlType | Label Type |
|  | IsMandatory | whether the control will be manadatory or not |
|  | ValidationType | Type of validation against the control |
|  | Maxlen | Text length in case of text, text area, html area |
|  | OrderID | the order in which the controls should get displayed |
|  | Minlen | This column contains limit size of input control |
|  | RadioButtonValue | stores radio button values like (Yes|No|NA) |
|  | AutoCaps | It stores a uppercase as a class which is for input text. |
|  | CustomValidation | It stores radio button values like (Yes|No|NA) |

**Columns Definition:**

DOCDID **:** This column is auto generated *means* primary key of the current table.

DOCMID **:** This column store the primary key or unique identifier of table (*WAT\_DynamicOutcomeControlMaster).*

DBColumnName: This columns are exists on table ( *WAT\_DataDetail\_ALG*) comes from table column form table *WAT\_DynamicOutcomeControlMaster.*

LabelName **:** It is the display name which is displayed on view page.

ControlType: This column contains control name form table (*WAT\_ControlTypeMaster*).

IsMandatory **:** This column holds 1 or 0 for mandatory or not.

ValidationType: Type of validation against the control like 1 for Isnum , 2 for isEmail, 3 for alfaNumeric.

Minlen : This column contains limit size of input control, if it is 0 or null then its max length will work.

RadioButtonValue: It stores radio button values like (Yes|No|NA)

AutoCaps: It stores a *uppercase* as a class which is for input text.

CustomValidation : It store special type of function defined by user/administrator. This function will work special or specific control which you want.

This is a special type of Column which contains a method which is used by any html control and its definition is provided by himself a/c to our need. If it implement then no any other validation will not work.

5. WAT\_DynamicDDLMst – This table maintains the master entries of Dropdown controls which are decided from the above table where Control Type is “1”.

Below are the table columns.

|  |  |  |
| --- | --- | --- |
| **TableName** | **ColumnName** | **Definition** |
| WAT\_DynamicDDLMst | DDLMID | Unique Column ID |
|  | DOCDID | Dynamic Control ID against the unique id of DDL (referenced from above table) |
|  | Name | Control Value for different control Ids |
|  | ParentDocDID | Dependent Id of control |
|  | IsDependent | Whether the control is dependent or not |
|  | ParentDocMID | Dependent control id |

**Here is an example to represent the data flow in the Dynamic Control Tables.**

This is the master entry in “WAT\_DynamicOutcomeControlMaster” table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DOCMID** | **WorkGMID** | **TableName** | **ClientID** | **FreezeStatus** |
| 8 | 1041 | WAT\_DataDetail\_ALG | 202 | 0 |

“WAT\_DynamicOutcomeControlMasterDetail” table we have entries corresponding to master entries which creates the controls on screen. DOCMID is referenced in this table for further relationships.

0-TextBox

1-Dropdown

2-TextArea

4- DataTime

7- Radio Button

OrderId keeps the controls in the ordered sequence.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DOCDID** | **DOCMID** | **DBColumnName** | **LabelName** | **ControlType** | **IsMandatory** | **ValidationType** | **Maxlen** | **FreezeStatus** | **OrderID** |
| 57 | 8 | Col1 | Case ID | 0 | 0 | 0 | 100 | 0 | 1 |
| 58 | 8 | Col2 | Claim Reference | 0 | 0 | 0 | 100 | 0 | 2 |
| 59 | 8 | Col3 | Contact Reference | 0 | 0 | 0 | 100 | 0 | 3 |
| 60 | 8 | Col4 | Primary Reason | 1 | 0 | 0 | 100 | 0 | 4 |
| 61 | 8 | Col5 | Secondary Reason | 1 | 0 | 0 | 100 | 0 | 5 |
| 62 | 8 | Col6 | Additional Notes 1 | 2 | 0 | 0 | 100 | 0 | 6 |

Now for control type “1” i.e. Dropdown, we keep the entries in “WAT\_DynamicDDLMst” table.

Corresponding of DOCDID 60, we have 3 records in below table. Since, it will be dependent control so it has IsDependent as 1.

ParentDocDID is 61 because all the data with DOCDID 61 will be reflected when the dropdown value with DDLMID as (39, 40 and 41) will be changed.

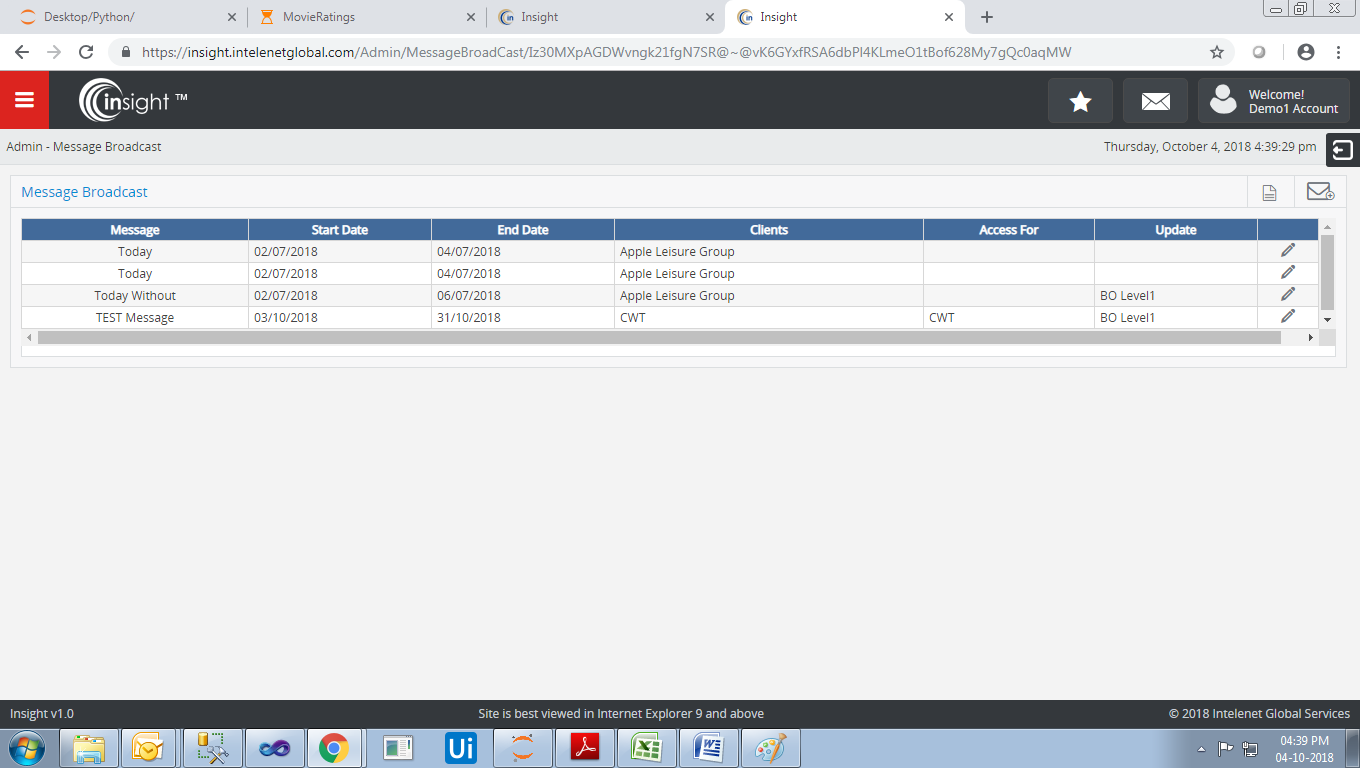
Now ParentDocMID keeps the DDLMID reference as if “Awaiting customer” is selected then we will fetch (46,47,48) DDLMIDs record.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DDLMID** | **DOCDID** | **Name** | **ParentDocDID** | **FreezeStatus** | **IsDependent** | **ParentDocMID** |
| 39 | 60 | Awaiting customer | 61 | 0 | 1 |  |
| 40 | 60 | Awaiting legal classification | 61 | 0 | 1 |  |
| 41 | 60 | Awaiting other department | 61 | 0 | 1 |  |
| 46 | 61 | Bank Details Requested | 0 | 0 |  | 39 |
| 47 | 61 | DPA Requested | 0 | 0 |  | 39 |
| 48 | 61 | Awaiting Consent | 0 | 0 |  | 39 |
| 49 | 61 | FDR Unconfirmed < 14 Days | 0 | 0 |  | 40 |
| 50 | 61 | FDR Not Available < 14 Days | 0 | 0 |  | 40 |
| 51 | 61 | FDR letter template missing < 14 Days | 0 | 0 |  | 40 |
| 52 | 61 | Airport Escalation – 1 | 0 | 0 |  | 41 |
| 53 | 61 | Airport Escalation – 2 | 0 | 0 |  | 41 |

**6. Message Broadcast Tables in WAT**

MessageBroadcast - Messgae notification to users

The data is saved in MessageBroadcast table.



**7. Report in WAT**

It will get created like proc\_rpt\_WAT\_( Report Name)

Ex. proc\_rpt\_WAT\_GeneralSummary

proc\_rpt\_WAT\_ProductivityDetail