Rule Engine - Master Data Management

Version 1.0

Symphony IPM 3.7

Approach Document

07-Dec-2021



MaxVal Group, Inc.

2251 Grant Road, Suite B

Los Altos, CA 94024

The content is a property of MaxVal Group Inc., USA. This content is for training, MaxVal employees and their clients. This should not be shared outside the organization, not should it be replicated in any form. For any queries on the content please contact info@maxval.com

## 1.Problem Statement

In the previous version (Symphony 1.5 and 3.6), the master data is uploaded from the spreadsheet through Data Loader or Data Import Wizard and also updated from another tool Annuity Payer.

## 1.1. Problems in spreadsheet data update

There are a lot of dependencies if an end user is uploading the data in each Symphony client. The dependencies are as shown below.

* Have to maintain the common data sheets offline such as google drive or any other file server.
* Have to maintain multiple copies of data sheets manually for the revisions.
* Have to do the changes directly in the data sheets as no UI provided to the rule owners and rule approvers.
* Have to upload the data sheets manually in Symphony clients and also we have to keep the track of changes manually

## 1.2. Problems in Annuity Payer connectivity

Some of the master data are fetched from Annuity Payer and those data are updated in Symphony client.

Dependencies are as shown below.

* Establish the connection between Annuity Payer and Symphony
* Changes will not be updated immediately as Symphony pulls the data from Annuity Payer on a daily basis.

## 2. Objective

The main objective is to resolve the above mentioned problems. The master data should be maintained in a centralized location which is ‘Rule Org’ and once the changes are done, then Rule Org should push all the master data changes to all the Symphony clients automatically or manually.

3. Proposed Approach

All the master data should be maintained in Rule org. The changes should be immediately reflected in the Symphony client once the changes happened in the master data.

## 3.1. Object Design for Master Data

Object: Jurisdiction

*Purpose*: To maintain all the available country details

**Field Name Data Type Remarks**

-------------------------------------------------------------------------------------------------------------------------------

Country Name Text (80) Name of the Jurisdiction/Country

Country Code Text (20) Unique.

Is Trademark Jurisdiction Checkbox To mark whether this field is used in Trademark module

PCT Contracting State Checkbox

EP Validation State Checkbox

Keywords Text (255) To save alias name for the country

Allowed Case Types <To be added>

Allowed Asset Types <To be added>

Allowed Entity Types

Is Active Checkbox True by default.

-------------------------------------------------------------------------------------------------------------------------------

Validations/Other Details:

* Country Code - Unique and Mandatory. This field will be the unique key to update the corresponding records in Symphony from Rule Org.
* Country Name - Mandatory.
* Is Active - This field will be marked as false if the record gets deleted from Rule Org.
* Keywords - This field is used to save the alias name of the country.
  + Eg: US, USA, United States for ‘United States of America’

Object: Currency

*Purpose*: To maintain all the available currency with its exchange rate details

**Field Name Data Type Remarks**

-------------------------------------------------------------------------------------------------------------------------------

Currency Name Text (80)

Currency Code Text (20) Unique

Exchange Rate Number (10,4)

Exchange Rate (Updated on) Date/Time

Country Lookup (Jurisdiction)

Is Active Checkbox True by default

-------------------------------------------------------------------------------------------------------------------------------

Validations/Other Details:

* Currency Code - Unique and Mandatory. This field will be the unique key to update the corresponding records in Symphony from Rule Org.
* Currency Name - Mandatory.
* Is Active - This field will be marked as false if the record gets deleted from Rule Org.

Object: Entity Type

*Purpose:* To maintain all the available entity types for each country. This will be used to calculate the renewal fee.

**Field Name Data Type Remarks**

-------------------------------------------------------------------------------------------------------------------------------

Entity Type Name Text (80)

Unique Id Number(10,0) Unique

Country Code Text (20)

Jurisdiction Lookup(Jurisdiction)

Is Default Checkbox

Is Active Checkbox True by default

-------------------------------------------------------------------------------------------------------------------------------

Validations/Other Details:

* Unique Id - Unique and Mandatory. Auto generated numbers should be stored in this field. This field will be the unique key to update the corresponding records in Symphony from Rule Org.
* Entity Type Name - Mandatory.
* Is Active - This field will be marked as false if the record gets deleted from Rule Org.
* Is Default - Each country can have only one default entity type record.
* Jurisdiction - Mandatory. Users should select the jurisdiction from the lookup object.
* Country Code - This field should be updated with the code if the Jurisdiction field is not empty when inserting or updating the record.

Object: Kindcode

*Purpose*: To maintain all the available kindcode for each country

**Field Name Data Type Remarks**

-------------------------------------------------------------------------------------------------------------------------------

Kindcode Name Text (80)

Unique Id Number (10,0) Unique

Country Code Text (20)

Jurisdiction Lookup (Jurisdiction)

Comments Text (255)

Status Text (255)

Is Active Checkbox True by default

-------------------------------------------------------------------------------------------------------------------------------

Validations/Other Details:

* Unique Id - Unique and Mandatory. Auto generated numbers should be stored in this field. This field will be the unique key to update the corresponding records in Symphony from Rule Org.
* Kindcode Name - Mandatory.
* Is Active - This field will be marked as false if the record gets deleted from Rule Org.
* Jurisdiction - Mandatory. Users should select the jurisdiction from the lookup object.
* Country Code - This field should be updated with the code if the Jurisdiction field is not empty when inserting or updating the record.

## 3.2. Functional Implementation for the master data objects

3.2.1. Jurisdiction Object

* Below fields should be displayed on the Jurisdiction UI to create new or edit the existing record.
  + Country Name, Country Code, Is Trademark Jurisdiction, PCT Contracting State, EP Validation State, Keywords, Is Active
* Country Code should be the unique key to find the corresponding record in Symphony client
* Country code’s values should also be unique
* Application should save the record only if there are changes done. Otherwise it should be intimate to the user with the below message.
  + ‘No changes done in the country’

3.2.2. Currency Object

* Below fields should be displayed on the currency UI to create new or edit the existing record.
  + Currency Name, Currency Code, Exchange Rate, Exchange Rate (Updated on), Is Active
* Currency Code should be the unique key to find the corresponding record in Symphony client
* Currency code’s values should also be unique
* Application should save the record only if there are changes done. Otherwise it should be intimate to the user with the below message.
  + ‘No changes done in the currency’

3.2.3. Entity Type Object

* Below fields should be displayed on the entity type UI to create new or edit the existing record.
  + Entity Type Name, unique id, Country Code, Jurisdiction, Is Default and Is Active
* Unique Id should be the unique key to find the corresponding record in Symphony client
* Unique Ids values should also be unique
* Entity Type Name + Jurisdiction - Unique
* ‘Is Default’ should be set for only one entity type per country
* Application should save the record only if there are changes done. Otherwise it should be intimate to the user with the below message.
  + ‘No changes done in the entity type’

3.2.4. Kindcode Object

* Below fields should be displayed on the entity type UI to create new or edit the existing record.
  + Kindcode, unique id, Country Code, Jurisdiction, comments, status and Is Active
* Unique Id should be the unique key to find the corresponding record in Symphony client
* Unique Ids values should also be unique
* Kindcode + Jurisdiction - Unique
* Application should save the record only if there are changes done. Otherwise it should be intimate to the user with the below message.
  + ‘No changes done in the kindcode’

## 3.3. Object Design for Transaction

Object: Client Release

*Purpose*: To maintain a queue for all the requests

**Field Name Data Type Remarks**

-------------------------------------------------------------------------------------------------------------------------------

Name Auto Number

Client Lookup (Client\_\_c)

Object Picklist (Country, Currency, Entity Type, KindCode, Docketing Rule)

Released by Lookup (User)

Released Date Date/Time

Status Picklist (Yet to Release, In Progress, Released, Failed, Archived, Canceled)

Message Text (32,768)

Response Content Text (32,768)

Attempt Number (4,0)

Pushtype Picklist (Automatic, Manual)

Requested Date Date/Time

Response Received Date Date/Time

-----------------------------------------------------------------------------------------------------------------------------

3.4. Push all the master data when a new client is created

This section explains how all the master data is pushed to the client when it is created and configured.

## 3.4.1. Flow Diagram



## 3.4.2. Implementation Steps

* Below mentioned objects should be pushed once new client is on boarded
  + Country, Currency, Entity Type, Kindcode, Docketing Rule
* Below records will be created in ‘Client Release’ once a new client record is created
  + Add a request in ‘Client Release’ to push country
    - Client Id = ‘Salesforce Id of the new client’
    - Object Name = ‘Country’
    - Status = ‘Yet to Release’
  + Add a request in ‘Client Release’ to push currency
    - Client Id = ‘Salesforce Id of the new client’
    - Object Name = ‘Currency’
    - Status = ‘Yet to Release’
  + Add a request in ‘Client Release’ to push entity type
    - Client Id = ‘Salesforce Id of the new client’
    - Object Name = ‘Entity Type’
    - Status = ‘Yet to Release’
  + Add a request in ‘Client Release’ to push kindcode
    - Client Id = ‘Salesforce Id of the new client’
    - Object Name = ‘Kindcode’
    - Status = ‘Yet to Release’
  + Add a request in ‘Client Release’ to push docketing rule
    - Client Id = ‘Salesforce Id of the new client’
    - Object Name = ‘Docketing Rule’
    - Status = ‘Yet to Release’
* A batch class to be created and it will be triggered when any changes done in queue
* Batch class will read all the pending requests from the push queue (‘Client Release’ object)
* Batch class will call the end point of each client to push the records for the specified objects in the request
* Track the response details in the same ‘Client Release’ object
* Status of the ‘Client Release’ should be changed to
  + ‘Released’ once all the records of the mentioned objects have been updated in the client.
  + ‘Failed’ if any error occurs on ‘Rule Org’ front or ‘Symphony’ front. Also, the error message will be saved in the ‘Message’ field.
  + ‘In Progress’ if the request is in process
  + ‘Cancelled’ if the user manually cancelled the push from the provided UI.
* A job will be scheduled to handle the failure cases
* That job will try a maximum of 5 attempts to push the ‘Failed’ cases. If unable to push, then that request will be ‘Archived’.
* Users have to verify the errors for the failed cases and it will be pushed once the issues get resolved either in the ‘Rule Org’ front or ‘Symphony’ front.
* One batch class will be executed to push master data to only one client at a time.
  + Country push batch will be executed first.
  + Currency push batch will be executed once the country push batch gets completed
  + Entity Type push batch will be executed once the currency push batch gets completed
  + Kindcode push batch will be executed once the entity type push batch gets completed

3.5. Push the master data to existing clients when it gets changed

This section explains how the master data changes are pushed to the client when it it gets changed or when a new record gets added

## 3.5.1. Flow Diagram



## 3.5.2. Implementation Steps

3.5.2.1. To add Jurisdiction in push queue

* Once the country record gets changed (inserted or updated), the application will automatically create a new record in the ‘Client Release’ object for all the available clients with the below fields and its values.
  + Client Id = ‘Salesforce Id of Client object’
  + Object Name = ‘Country’
  + Status = ‘Yet to Release’

Note: 25 ‘Client Release’ records will be created if there are 25 Symphony clients.

3.5.2.2. To add currency in push queue

* Once the currency record gets changed (inserted or updated), the application will automatically create a new record in the ‘Client Release’ object for all the available clients with the below fields and its values.
  + Client Id = ‘Salesforce Id of Client object’
  + Object Name = ‘Currency’
  + Status = ‘Yet to Release’

Note: 25 ‘Client Release’ records will be created if there are 25 Symphony clients.

3.5.2.3. To add entity type in push queue

* Once the entity type record gets changed (inserted or updated), the application will automatically create a new record in the ‘Client Release’ object for all the available clients with the below fields and its values.
  + Client Id = ‘Salesforce Id of Client object’
  + Object Name = ‘Entity Type’
  + Status = ‘Yet to Release’

Note: 25 ‘Client Release’ records will be created if there are 25 Symphony clients.

3.5.2.4. To add entity type in push queue

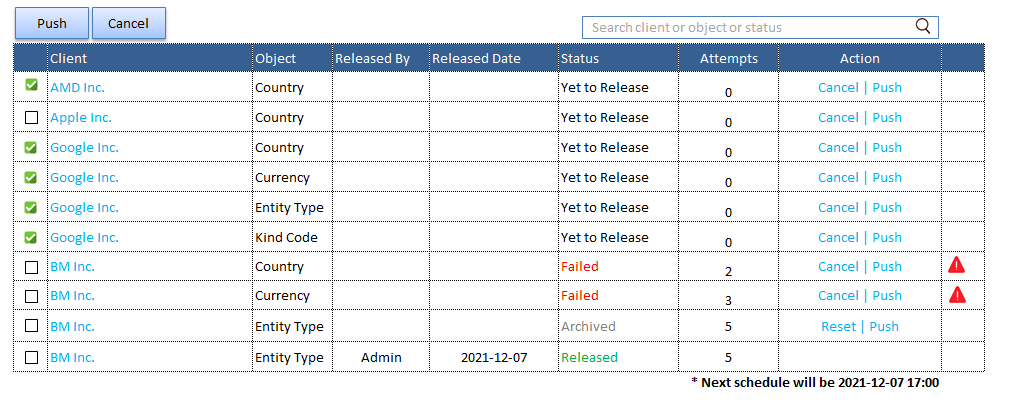
* Once the entity type record gets changed (inserted or updated), the application will automatically create a new record in the ‘Client Release’ object for all the available clients with the below fields and its values.
  + Client Id = ‘Salesforce Id of Client object’
  + Object Name = ‘Kindcode’
  + Status = ‘Yet to Release’

Note: 25 ‘Client Release’ records will be created if there are 25 Symphony clients.

## 3.5.2.5. Batch class to read the push request from the queue

* A batch class to be created and it will be triggered when any changes done in queue
* Batch class will read all the pending requests from the push queue (‘Client Release’ object)
* Batch class will call the end point of each client to push the records for the specified objects in the request
* Track the response details in the same ‘Client Release’ object
* Status of the ‘Client Release’ should be changed to
  + ‘Released’ once all the records of the mentioned objects have been updated in the client.
  + ‘Failed’ if any error occurs on ‘Rule Org’ front or ‘Symphony’ front. Also, the error message will be saved in the ‘Message’ field.
  + ‘In Progress’ if the request is in process
  + ‘Cancelled’ if the user manually cancelled the push from the provided UI.
* A job will be scheduled to handle the failure cases
* That job will try a maximum of 5 attempts to push the ‘Failed’ cases. If unable to push, then that request will be ‘Archived’.
* Users have to verify the errors for the failed cases and it will be pushed once the issues get resolved either in the ‘Rule Org’ front or ‘Symphony’ front.

## 3.6. UI to view the ‘Push Queue’ (Salesforce platform page can be used without few options)



* User can do the below activities
  + sort the list by Client, object, Status
  + Search the result by client name or object or status
  + See the detail view of each request
  + Select bulk requests and can make action such as ‘Push’ or ‘Cancel’
* Users don't need to wait for the next schedule. They can push the data manually to the selected clients.

## 4.0. Removal of Annuity Payer connectivity

| **Annuity Payer**  **Objects** | **Symphony 3.7** | **Remarks** |
| --- | --- | --- |
| Country | Country | Use the same object as Master Data |
| Currency | Currency | Use the same object as Master Data with Currency exchange rate |
| Patent Status | Not Required | Replaced with Patent’s Status field |
| Patent Type | Not Required | Replaced with Patent’s Case Type field |
| Entity Type | Entity Type | Use the same object as Master Data |
| Term Rule | Not Required | Replaced with Docketing Rule |
| Fee | Not Required | Replaced with Docketing Rule |
| Commission | Not Required | MaRS provides this data |
| Agent Charge | Not Required | MaRS provides this data |
| Markup Percentage | Not Required | MaRS provides this data |
| Expiry Rule | Not Required | Replaced with Docketing Rule |