Technical Specifications

Facets Batch Framework - Technical Specifications

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prepared By | Reviewed By | Approved By |
| Name | Himanshu S Srivastava |  |  |
| Signature | Himanshu |  |  |
| Date | 03/01/2017 |  |  |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer |
| 03/03/2017 | 1.0 | Initial Version | Himanshu |  |
| 02/26/2018 | 1.1 | Upated for CLMU configuration | Raghav Sharma |  |

Contents

[1. Overview 3](#_Toc423519930)

[1.1. Purpose 3](#_Toc423519931)

[1.2. Functional Description 3](#_Toc423519932)

[1.3. Related Application Requirements 3](#_Toc423519933)

[1.4. Key Considerations 3](#_Toc423519934)

[1.1. Related Documents References 4](#_Toc423519935)

[1.2. Acronyms and Terms 4](#_Toc423519936)

[2. Technical Description 5](#_Toc423519937)

[2.1. Processing Logic 5](#_Toc423519938)

[2.2. Batch Files 11](#_Toc423519939)

[2.3. Restart Processing 11](#_Toc423519940)

[2.4. Error Processing 11](#_Toc423519941)

[2.5. Architectural Mechanism 11](#_Toc423519942)

[2.6. Application Interfaces 11](#_Toc423519943)

[2.7. Security 11](#_Toc423519944)

[3. Run Control Execution 12](#_Toc423519945)

# Overview

## Purpose

This document addresses the overview of Facets Batch Framework and the steps required to use the common bacth framework to execute any Facets batch.

## Functional Description

This main function of this framework is the processing of Facets core batches using common batch wrapper NT\_COMMON\_BATCHWRPR . The common wrapper executes a custom batch wrapper To read the configuration xml,perform any pre/post processes, modify the run book and execute the batch.

This script receives batch name and database name as parameters. Based on the batch name, the corresponding section in the configuration file is read and all the batch specific information are fetched and stored in a global dictionary object. Also, the generic information are fetched and stored in the dictionary object.

The various steps in the pre process xml are read and the required steps are executed. The runbook for the batch is modified with the override parameters and the modified runbook is run to execute the batch. The various steps in the post process xml are read and the required steps are executed.

## Assumptions

1. The configuration file "commonsystemprop.xml" is available in the application path.
2. The VB DLL JD\_NTBATCH\_CMNFUNCS is registered in the system registry.
3. The folders configured in the configuration file "commonsystemprop.xml" are available.
4. The executable Blat.exe used to send mail is available in the path mentioned in the commonsystemprop.xml.
5. The input files for the batches are named according to the naming convention mentioned in the configuration file.

## Related Application Requirements

|  |  |
| --- | --- |
| # | Other Programs / Modules / Interfaces that will help to meet this requirement |
| 1 | JD\_NTBATCH\_CMNFUNCS.dll |
|  |  |

## Key Considerations

*NA*

|  |  |  |
| --- | --- | --- |
| Impact (New, Enhancement) | Object/Module/Program/File Type | Object/Module/Program/File Name |
|  |  |  |
|  |  |  |
|  |  |  |

## Related Documents References

This document use references from the following artifacts. <Any business document that can be pulled out from ADR\ SharePoint\Rally>

| Lockbox Functional document |
| --- |
|  |

## Acronyms and Terms

The following list of acronyms and terms are used within this document:

| Acronym / Term | Description |
| --- | --- |
|  |  |

# Technical Description

## Processing Logic

Add configuration in commonsystemprop.xml and provide batch name under tag <Category name>, add all generic configurations such as Input/Output dir.In OverrideParameters add all batch specific configurations :

<Category name="HADC\_CMCBRCP\_BILL\_RECEIPTS\_FISERV">

<Item name="Mail\_to\_ok">email@Optum.com</Item>

<Item name="Subject">Log for Billing Receipts batch</Item>

<Item name="Body"></Item>

<Item name="Mail\_to\_err">email@Optum.com</Item>

<Item name="InputDir">E:\Batch\_Framework\FTP\BL\_IN</Item>

<Item name="OutputDir">E:\Batch\_Framework\FTP\BL\_OUT</Item>

<Item name="ProductAppId">BRCP</Item>

<Item name="ProductName">FA</Item>

<Item name="OverrideParameters">

<Element name="RunBookName">ercmcrunbrcp.xml</Element>

<Element name="InputFile">Config("InputDir") + billing.receipt.payment.fiserv</Element>

<Element name="InputFileExtension">txt</Element>

<Element name="OutputFile">Config("OutputDir") + billing.receipt.payment.fiserv</Element>

<Element name="RunDate">mm/dd/yyyy 00:00:00.000</Element>

<Element name="BatchId"></Element>

<Element name="HipaaIndicator">N</Element>

<Element name="HipaaAltIndexFile"></Element>

<Element name="HipaaBaseSystemInstance"></Element>

<Element name="HipaaRepositoryInstance"></Element>

</Item>

</Category>

Add the input files as per the configured InputDir and make sure that the input files are named according to the naming convention mentioned in the configuration file.

<Begin Version 1.1>

<Category name="CMCRUN\_CLMU">

<Item name="Mail\_to\_ok">FacetsClaimsSupport</Item>

<Item name="Subject">Log for Medical Electronic Adjudication - ErCmcRunClmu</Item>

<Item name="Body"></Item>

<Item name="Mail\_to\_err">FacetsClaimsSupport</Item>

<Item name="ProductAppId">CLMU</Item>

<Item name="ProductName">FA</Item>

<Item name="Pre-Process">

<Element name="Xml">PreProc\_Clmu.xml</Element>

</Item>

<Item name="Post-Process">

<Element name="Xml">PostProc\_Clmu.xml</Element>

</Item>

<Item name="OverrideParameters">

<Element name="RunBookName">ercmcrunclmu\_test.xml</Element>

<Element name="DeleteXcClaims">N</Element>

<Element name="RunDate">mm/dd/yyyy 00:00:01.001</Element>

<Element name="NumberOfEnginesClmu">2</Element>

<Element name="NumberOfQuesClmu">6</Element>

<Element name="WorkflowReprocess">Y</Element>

<Element name="DeadlockRetryAttempts">3</Element>

<Element name="DeadlockRetryWaitSeconds">3</Element>

<Element name="ApplicationPzapAppId">EADJ</Element>

</Item>

</Category>

<!--

#################################################################

# Title : PreProc\_Clmu.xml

# Purpose: Pre-Process Steps for Medical Electronic Adjudication Batch

# Log :

# Initial revision.

#################################################################

-->

<Job>

<Category name="Indicative">

<Item name="JobName">PreProc\_Clmu</Item>

<Item name="JobDescription">Pre-Process Electronic Adjudication Procedure Book</Item>

</Category>

<!--

#################################################################

# Step 1000

#################################################################

-->

<Step number="1000">

<Category name="Indicative">

<Item name="StepName">PreProc\_Clmu\_1000</Item>

<Item name="StepDescription">Update the run date in the Override Dictionary Object</Item>

<Item name="StepNumber">1000</Item>

</Category>

<!--

#################################################################

# Action 1000\_1000

#################################################################

-->

<Action number="1000\_1000">

<Category name="Indicative">

<Item name="ActionNumber">1000\_1000</Item>

<Item name="ProgramName">DLL</Item>

</Category>

<FunctionInfo>

<Function>UpdateDicOverride</Function>

<Parameter>

<Item>

<Name>Item</Name>

<Value>RunDate</Value>

</Item>

</Parameter>

</FunctionInfo>

</Action>

</Step>

</Job>

<!--

#################################################################

# Title : PostProc\_Clmu.xml

# Purpose: Post-Process Steps for Batch Billing Receipt Processing

# $Log:

# Initial revision.

#################################################################

-->

<Job>

<Category name="Indicative">

<Item name="JobName">PostProc\_Clmu</Item>

<Item name="JobDescription">Post-process Electronic Adjudication Procedure Book</Item>

</Category>

##################################################################

# Step 1000

##################################################################

-->

<Step number="1000">

<Category name="Indicative">

<Item name="StepName">PostProc\_Clmu\_1000</Item>

<Item name="StepDescription">Identify the log file name for the batch</Item>

<Item name="StepNumber">1000</Item>

</Category>

<!--

##################################################################

# Action 1000\_1000

##################################################################

-->

<Action number="1000\_1000">

<Category name="Indicative">

<Item name="ActionNumber">1000\_1000</Item>

<Item name="ProgramName">DLL</Item>

</Category>

<FunctionInfo>

<Function>GetLogFileName</Function>

<Parameter>

<Item>

<Name>ProductName</Name>

<Value>Config("ProductName")</Value>

</Item>

<Item>

<Name>BatchName</Name>

<Value>Config("ProductAppId")</Value>

</Item>

</Parameter>

</FunctionInfo>

</Action>

</Step>

</Job>

<End Version 1.1>

Call NT\_COMMON\_BATCHWRPR script using batch name and database name as parameters. Based on the batch name, the corresponding section in the configuration file is read and all the batch specific information are fetched and stored in a global dictionary object. Also, the generic information are fetched and stored in the dictionary object.

Once batch completed go into C:\Batch\_Framework\Log to check logs.

## Batch Files

## Restart Processing

* + 1. Program Restart

*<****TBD****: Describe commit/restart-processing logic. Include commit checkpoints and any synchronization that is required to restart the program from the last commit >*

* + 1. Batch Restart

There is no specific processing or logic to be applied to restart the job. The job can be triggered again from start if it fails at any of the steps. The job will start processing the remaining data.

## Error Processing

*<Describe error processing logic here including pseudo-code, input/output table details (Table Name, Field, Field Size, and Field Format), new screens, reports, error codes and error descriptions, etc.>*

## Architectural Mechanism

*<Describe interdependencies and interactions with other programs, modules, events, and interfaces.>*

## Application Interfaces

*<If this Technical Specification is for a new application interface or changes to an existing application interface, complete the Application Interface Specification Template defined in UDP.>*

## Security

<NA>

# Run Control Execution

This will be a Daily job and will run in sequence with the Claim Load process. More details on the execution environment will be added as and we have more clarity on the execution environment.