**ExxonMobil Chemical Company**

**Vistalon Application**

**Design Specification**

**Version 1.5**

**March 11, 2016**

**Prepared by:**

**ILS Automation Inc.**

REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Initial** | **Description** |
| 1.0 | June 30, 2014 | Pete Hassler | First Draft |
| 1.1 | May 10, 2015 | Pete Hassler | Updated recipe section and added unit configuration and lab data selection section |
| 1.2 | August 8, 2015 | Pete Hassler | Added SFC discussion and startup description |
| 1.3 | September 11, 2015 | Pete Hassler | Added discussion of reactor configuration design |
| 1.4 | September 27, 2015 | Pete Hassler | Fixed typo in section 6.2 |
| 1.5 | March 11, 2016 | Pete Hassler | Updated site specific tags in section 6.1, added section 7.2.1, mlr() derived values. |

**Table Of Contents**

1 Introduction 5

2 System Interfaces 5

2.1 OPC Interfaces 5

2.2 OPC HDA Interfaces 5

2.3 Database Interfaces 6

2.4 Tag History 8

3 Custom Console 9

4 Startup Scripts 10

4.1 Gateway 11

4.2 Client 12

5 Recipe Data 13

5.1 Recipe Control Screen 13

5.2 Recipe Tag Folder Structure 14

5.3 Local Recipe Tags 14

5.3.1 Local RLA3 Tags 15

5.3.2 Local RLA3 Tower Tags 15

5.3.3 Local VFU Tags 15

6 Grade Change and Reactor Configuration Handling 15

6.1 Grade Change 16

6.2 Reactor Configuration & Lab Data Selectors 16

7 Lab Data Facilities 19

7.1 Custom Validation Procedures 19

7.2 Custom Derived Values 19

8 SFC Migration 19

8.1 Re-engineered Diagrams 19

8.1.1 Moisture Removal 20

8.1.2 Extended Heat Soak 21

8.1.3 Yes No Choice with Timeout 21

9 Common Facilities 21

9.1 User Mode / User Group 21

9.2 Single Sign-On 22

# Introduction

This document discusses site specific configuration and customization for the new Ignition based Graphical Toolkit applications. It does not discuss the individual toolkit designs, which are covered in their own respective documents.

# System Interfaces

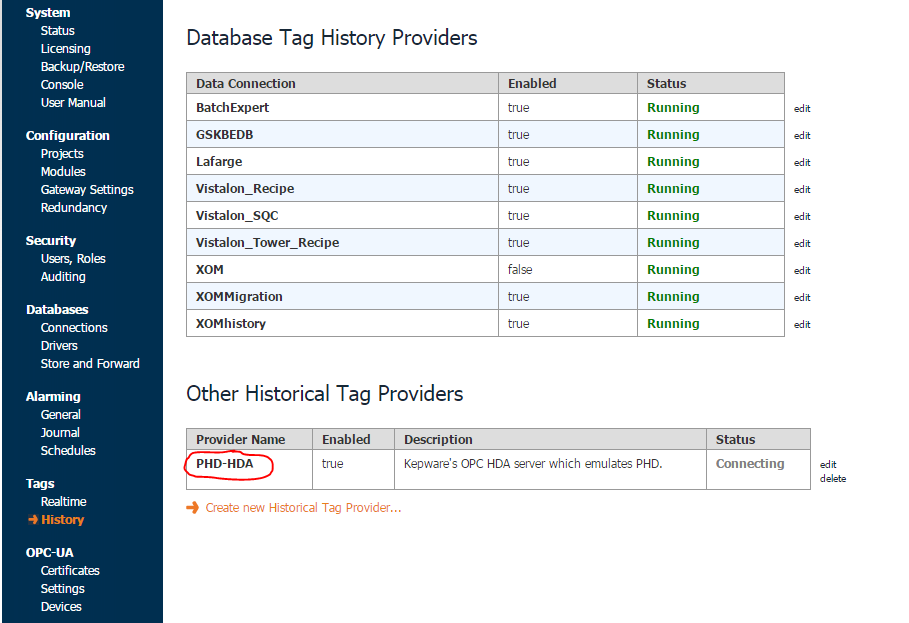
The following interfaces to external systems needs to be defined.

## OPC Interfaces

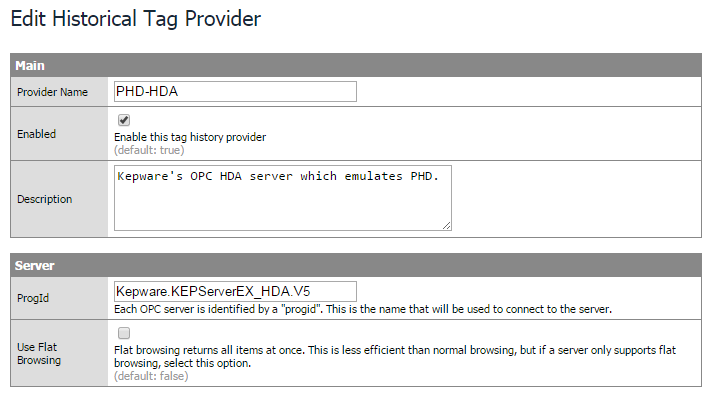
The Vistalon application serves the RLA3 console and the VFU console. Therefore there are two custom consoles:

## OPC HDA Interfaces

The Vistalon application serves the RLA3 console and the VFU console. Therefore there are two custom consoles:



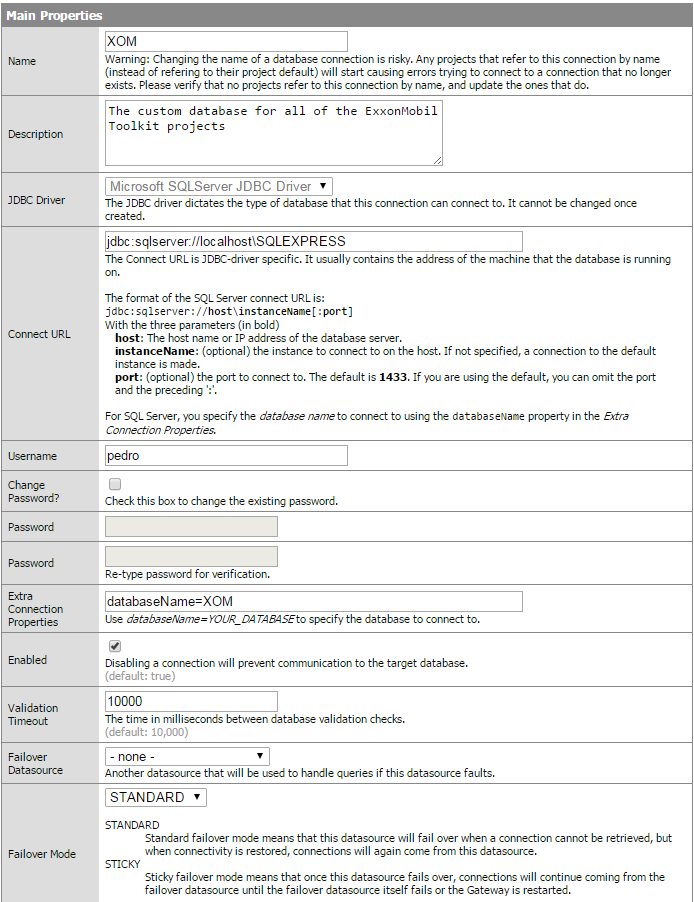
The configuration is shown below, the ProgId field should be changed to use you OPC HDA configuration:



## Database Interfaces

The application requires two database interfaces: XOM and XOMhistory. The XOM database houses all of the custom database tables designed for the application including the recipe tables which were formerly in a MS Access database. The XOMhistory database is used for storing tag history. Both interfaces are MS SQL\*Server.

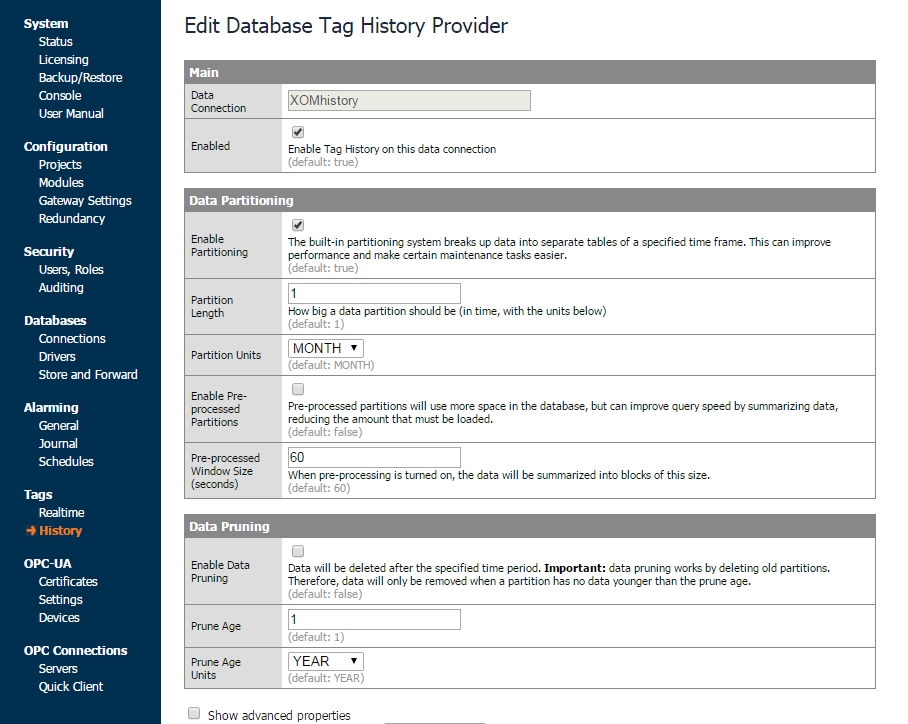
The XOM database configuration is similar to the following, the connect URL, username, and password should be modified for the site:



The XOMhistory connection is the same except for the “databaseName=XOMhistory” setting in the *Extra Connection Properties* field.

## Tag History

The amount of history that is stored in the Ignition tag historian is also configured from the Ignition Gateway web page under Configure::Tags::History. The configuration is shown below:



# Custom Console

The Vistalon application serves the RLA3 console and the VFU console. Therefore there are two custom consoles:

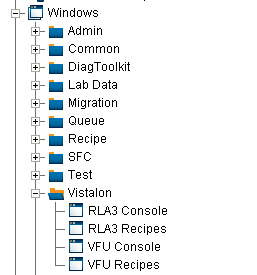


Figure - Vistalon Specific Windows

The RLA3 console is shown below:

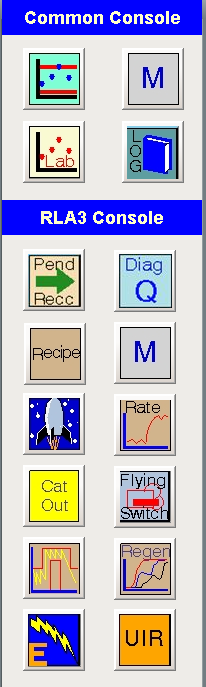


Figure - RLA3 Console

While the console and buttons are somewhat generic and reusable, they are typically configured to pass the post.

The console that is opened is based on the post. If the username that logs in matches the name of a post, then all of the windows that are defined in the database table TkConsole for that post are opened.

# Startup Scripts

This section describes the gateway and client startup scripts which are always sites specific but must contain some common elements.

## Gateway

The gateway startup script is shown below. It performs: 1) gateway related startup tasks for each of the toolkits, and 2) calls a custom site specific script.

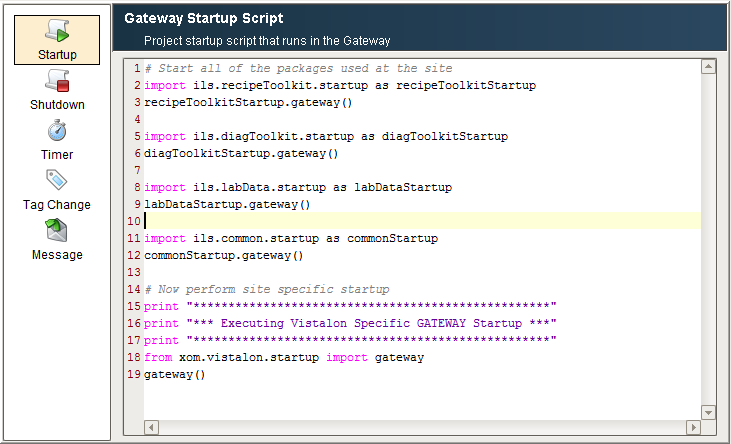
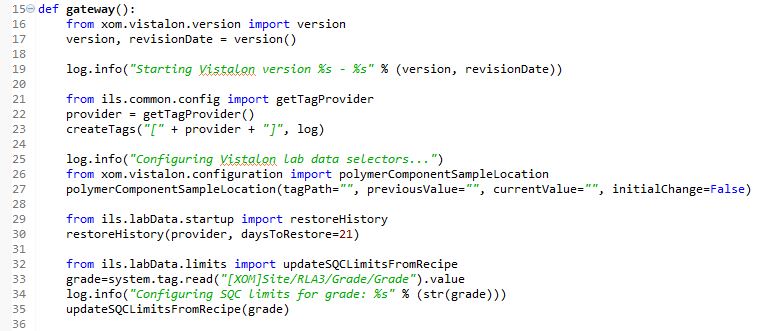


Figure - Site Specific Gateway Startup Script

The site specific startup script is shown below. It performs:

1. Creating site specific tags.
2. Configuring lab data selectors.
3. Restoring lab data history.
4. Updating SQC limits from recipe for the current grade



## Client

The client startup script is shown below. It performs: 1) client related startup tasks for each of the toolkits, 2) opens screens assigned to this post (*console.startup*), and 3) calls a custom site specific script.

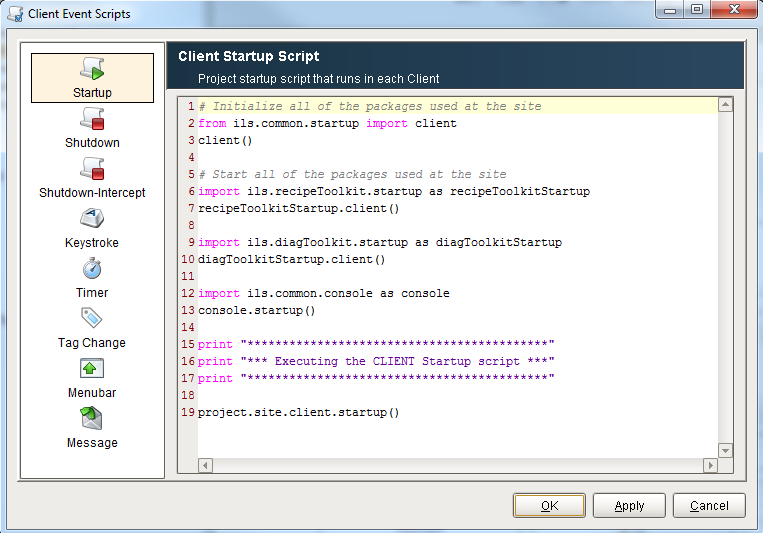


Figure - Client Startup Script

# Site Specific Tags

This section provides a description of the tags that are manually configured. It does not include tags that are automatically created by exporting configuration from G2 for the recipe, lab data, diagnostic toolkit, or SFC toolkits.

|  |  |  |  |
| --- | --- | --- | --- |
| Folder | Tag | Tag Type | Description |
| RLA3 | POLYSPLIT | OPC Float | Used by the derived variable calculations for RX2-C2-LAB-DATA and RX2-C9-LAB-DATA. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Recipe Data

This section provides a description of the operator console and user interface.

## Recipe Control Screen

A custom recipe selector screen exists for RLA and VFU. This is the gold screen shown below.



Figure - Recipe Screens

The buttons on the gold screen are configured with the recipe *familyName* as shown below:

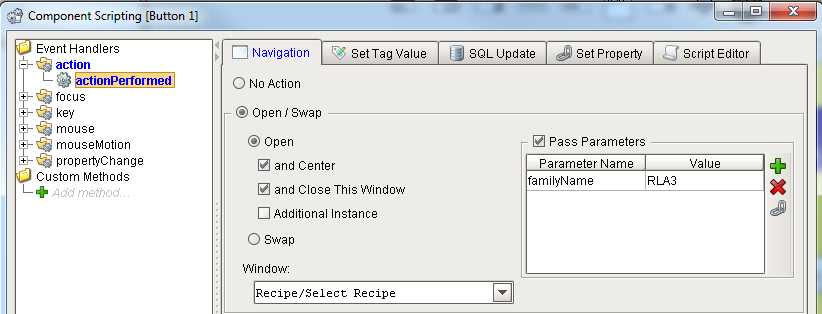


Figure - Recipe Screen Value Passing

## Recipe Tag Folder Structure

The folder structure for recipe data tags is shown below. Tags are created and deleted from the appropriate folder automatically by the recipe toolkit.

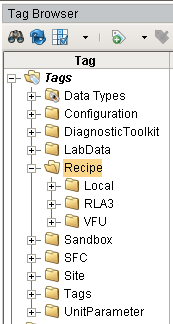


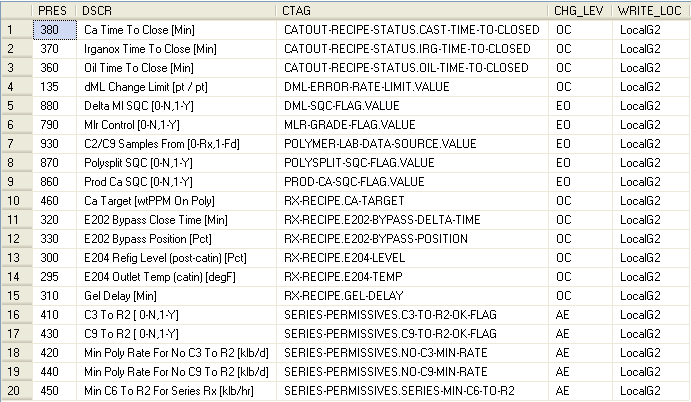
Figure 7 – Recipe Tag Folder Structure

## Local Recipe Tags

The Vistalon application has three recipes databases that require the following “local” tags. Local tags are updated from the recipe database, are not sent to the DCS, but are used by other graphical toolkits. These tags are created manually, either by hand or via a script, and are updated as part of the recipe download.

### Local RLA3 Tags

The RLA3 recipe has the following local tags:

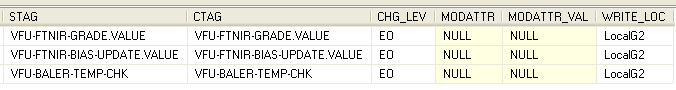


### Local RLA3 Tower Tags

There are no local tags for this recipe.

### Local VFU Tags

The VFU recipe has the following local tags:





# Grade Change and Reactor Configuration Handling

This section provides a description of the site specific grade change and reactor configuration handing capabilities.

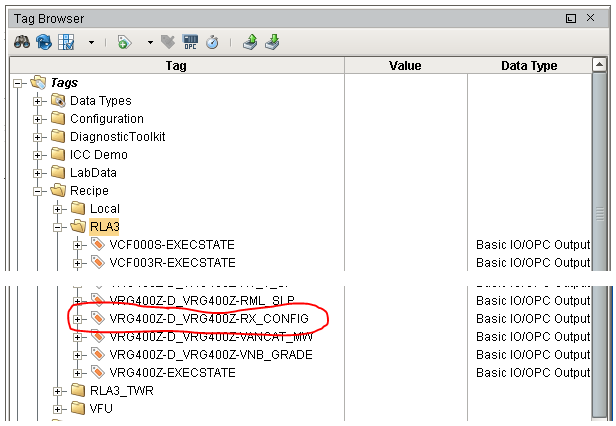
## Grade Change

In addition to a grade tag in recipe, which is written to during a recipe download, there is an instance of a grade UDT in the *Site/CRX* and in the *Site/VFU* folders. The grade UDT is used for reading the current grade and the grade IO UDT in recipe is used for writing the grade. The Grade UDT

## Reactor Configuration & Lab Data Selectors

A change in the reactor configuration needs to trigger a change in lab data selectors. The reactor configuration is ultimately set from a recipe download but the trigger for this logic is read from the DCS.

The recipe download toolkit write to the tag circled below.



There is an OPC tag, circled below, that reads from the DCS, this ensures that the recipe download was received by the DCS. This tag has a change script, *xom.vistalon.configuration.rxConfigDecoder*, which is called when it receives a new value.

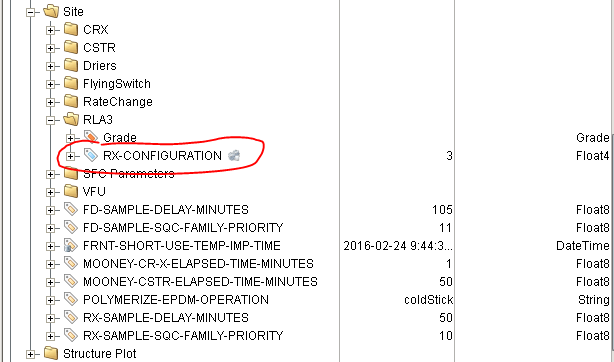
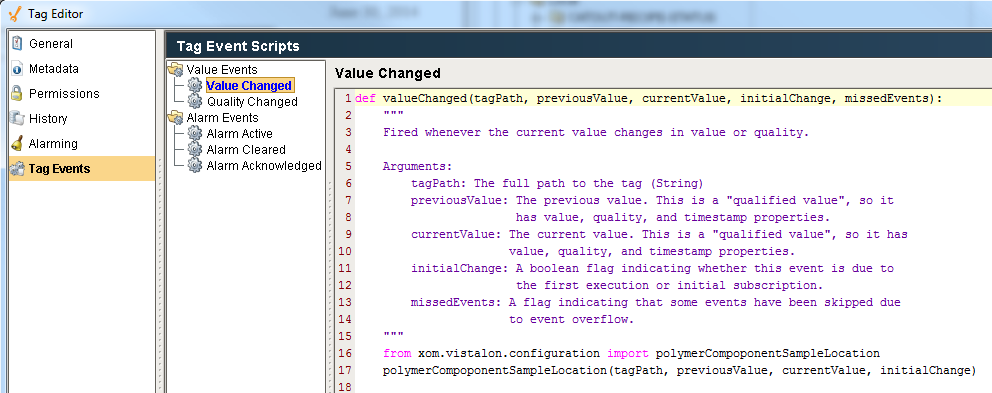


Figure 8 - Vistalon Configuration Tags

It decodes the integer reactor configuration value into the individual Boolean memory circled in red below. These four tags plus the POLY-PROP-FROM-FD circled in blue, which comes directly from recipe, are used to configure the lab data selectors. Two of the tags have event handlers defined that perform the selector reconfiguration. The tags are ***SERIES-RX-GRADE*** and ***POLY-PROP-FROM-FD*** as shown below.



The icon to the right of the tag name indicates there is an event handler defined on the tag, as shown below:



The script that is called is implemented in external Python in the XOM area.

There are a number of configuration tags that are used when the reactor configuration changes. The parameters from the old application are shown below:



These tags are automatically created on startup and their default values set in *xom.vistalon.startup.py*. The Ignition tags are shown in Figure 8.

Note: The local recipe tags are generally created programmatically via a startup script. This works pretty well until I wanted to put scripts on some of them. So in the future, perhaps it doesn’t make sense to make them programmatically because then you can end up with a complete looking set of tags that don’t have the scripting functions. It is more obvious something is wrong if you don’t have any tags.

# Lab Data Facilities

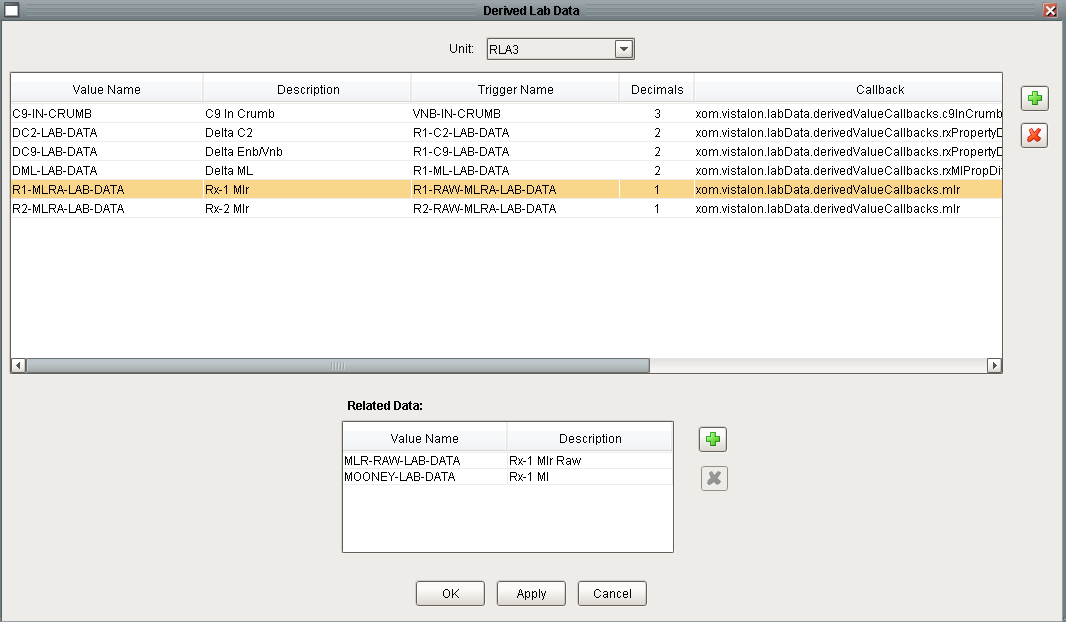
## Custom Validation Procedures

There are a number of lab data tags defined for the VFU that utilize a custom validation callback.

## Custom Derived Values

There are a number of lab data tags defined for the VFU that utilize a custom validation callback.

### Mlr

There are two derived variables, R1-MLRA-LAB-DATA and R2-MLRA-LAB-DATA, which use the same calculation method, mlr(), as shown below. 

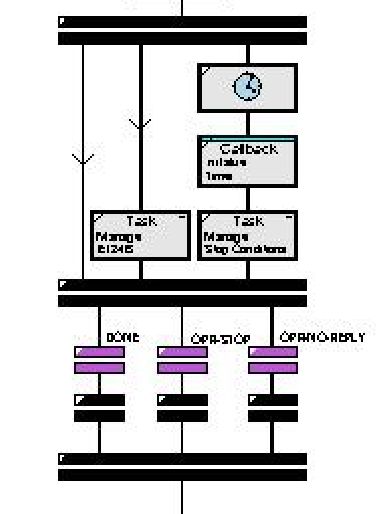
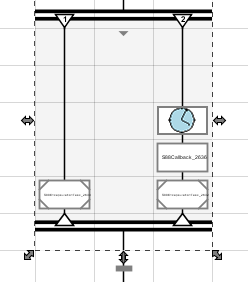
The trigger values are R1-RAW-MLRA-LAB-DATA and R2-RAW-MLRA-LAB-DATA respectively. The related data for moth of these derived variables are the same, MLR-RAW-LAB-DATA and MOONEY-LAB-DATA both of which are selectors for their respective R1 and R2 values. The two trigger tags are the source for the MLR-RAW-LAB-DATA related data tag. The trigger values are not referenced in the calculation method because the related tag is the same value. The calculation method references two additional items: MLR-SLOPE, which is an OPC tag read from the DCS, and MST-BASIS, which it calculates from the recipe database for the current grade. The recipe database is the source for the basis, because the limits currently in effect in the DCS could have been edited by the engineer.

# SFC Migration

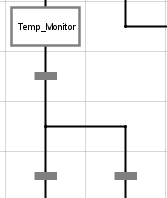
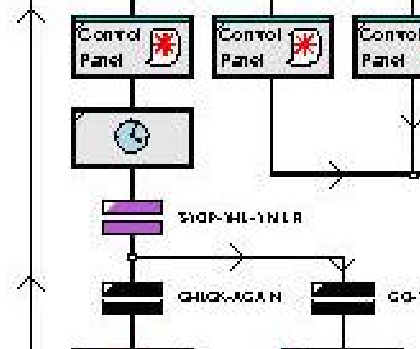
## Re-engineered Diagrams

Several SFC diagrams contained aspects in the old system that required re-engineering when converted to the new system. The functionality of these aspects should remain the same, but the appearance may be different.

### Moisture Removal

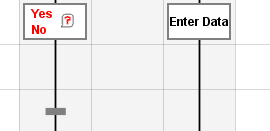
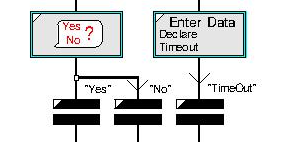
 The old system required two sets of parallel blocks while the new requires only one and for every parallel block the straight connection has been removed. Termination transition blocks of the old system (purple transition blocks) have been implemented into the cancel conditions of the parallel block in the new system. Finally, the three transition blocks underneath the termination transition blocks of the old system all performed the same task. The new system does not require this redundancy as these blocks were converted to a single transition outside of the parallel block.

### Extended Heat Soak



Since termination transitions do not exist in the new system, the combination of the delay block and the termination transition were implemented as the “Temp\_Monitor” and transition block shown in the diagram on the right. An “On Timer” function, set to run every second, determines if the termination conditions have been met. Once met, or upon timeout, the transition block allows the chart to proceed.

### Yes No Choice with Timeout



The three transition blocks in the original system were combined into a single transition block under the new system that has multiple conditions.

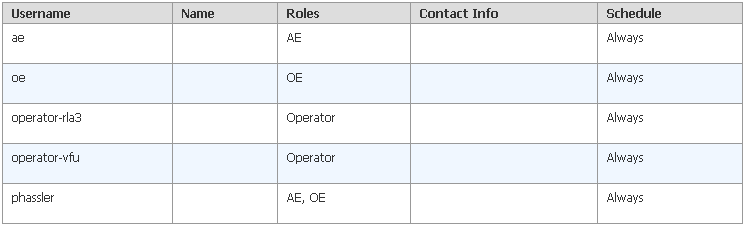
# Common Facilities

## User Mode / User Group

The details of system access / system security have not been determined. But the Recipe Toolkit was designed to distinguish between three roles:

|  |  |
| --- | --- |
| Role | Description |
| Operator | An operator. Cannot see OE recipe data and cannot edit AE recipe data. |
| AE | Can edit all recipe data |
| OE | Cannot access the recipe toolkit, but can access the recipe database application |

The following accounts have been configured on the ILS Automation development system to demonstrate role-based capabilities.



## Single Sign-On

Ignition supports Windows Active Directory and Single Sign On which means that once an operator logs ion to their workstation, they will not need to log on to Ignition. The credentials used to log on to the workstation will be used by Ignition. This will require the assistance of ExxonMobil’s systems group to set up the Windows groups to correspond with the roles defined above.