Integration Framework – Using Integration Processing Rules to customize your integration processing.

Introduction

The Integration Framework supports the use of Integration Controls and Processing Rules to allow customers to control and customize their integration transaction processing through Configuration. This document provides examples on how Processing Rules can be configured to customize integration message processing.

Table of Contents

Overview........................................................................................................................................... 2
Processing Rules/Integration Controls – Example data............................................................... 3
Processing Rule Definition ............................................................................................................ 9

NOTE: this document is a work-in-progress and will be updated to include more examples of Processing Rule configuration (Last updated - July 2013).
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Overview

The Maximo* Integration Framework (MIF) provides the capability to configure integration controls and processing rules to allow customers to customize their integration processing without (or in addition to) using custom Java User Exit code. Some the common uses of these features is to Set field value in integration or to filter integration messages such that only certain integration are sent and/or received.

This document is intended to walk you through multiple examples, in detail, of how you can take advantage of this capability within Maximo. The example screen shots included in this document are from a Maximo 7.5.x environment, however, the majority of the functionality is also available in the Maximo 7.1.x versions as well.

*NOTE: All capabilities provided in the Maximo application also apply the Smart Cloud Control Desk (SCCD) application
Integration Framework – Using Integration Processing Rules to customize your integration processing.

**Processing Rules/Integration Controls – Example data**

This section of the document will walk through multiple examples of the use of Integration Controls and Processing Rules. To support the examples used, the following data was configured.

An object structure ASSET2 was created with the Asset and Assetspec objects included. Additionally, only a selected number of attributes for each object were included (example XML is further down).
Integration Framework – Using Integration Processing Rules to customize your integration processing.

A new Publish Channel, Asset2Channel, was configured to use ASSET2 object structure.

The Enable Listener was turned on so that an update of the asset object would trigger and outbound message through this channel.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

This Channel was associated to two external systems, ExtSysA and ExtSysB, and the end point for each is set to an XML file.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Using this Channel a Data Export was done using ExtSysA and the XML file, ExtSysA_Asset2Channel_134391191117338630.xml, was created. Below is the content of that file. The asset exported was 11450 and it contained the following specification attributes (assetspec): TTLHD, STAGE, SPEED, SIZE, NPSH, DRIVER, CAPACITY,

```xml
<?xml version="1.0" encoding="UTF-8"?>
<PublishASSET2 xmlns="http://www.ibm.com/maximo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" creationDateTime="2012-08-02T08:51:50-04:00" transLanguage="EN" baseLanguage="EN"
messageID="1343911911173392936" maximoVersion="7 5 20120605-2230 V7503-156"
event="0">
  <ASSET2Set>
    <ASSET>
      <ASSETID>35</ASSETID>
      <ASSETNUM>11450</ASSETNUM>
      <ASSETTYPE />
      <CHANGEDATE>2005-02-09T08:24:53-05:00</CHANGEDATE>
      <DEFAULTREPFACSITEID />
      <DESCRIPTION>Centrifugal Pump 100GPM/60FTHD</DESCRIPTION>
      <DISABLED>0</DISABLED>
      <HIERARCHYPATH>PUMP \ CNTRFGL</HIERARCHYPATH>
      <INSTALLDATE>1996-05-26T00:00:00-04:00</INSTALLDATE>
      <LOCATION>BR450</LOCATION>
      <MANUFACTURER>IR</MANUFACTURER>
      <ORGID>EAGLENA</ORGID>
      <PARENT>11400</PARENT>
      <SERIALNUM>377-9B</SERIALNUM>
      <SITEID>BEDFORD</SITEID>
      <STATUS maxvalue="NOT READY">NOT READY</STATUS>
      <STATUSDATE>1998-09-16T17:58:00-04:00</STATUSDATE>
      <ASSETSPEC>
        <ASSETATTRID>TTLHD</ASSETATTRID>
        <CLASSSTRUCTUREID>1001</CLASSSTRUCTUREID>
        <LINEARASSETSPECID>0</LINEARASSETSPECID>
        <NUMVALUE>60.0</NUMVALUE>
        <ORGID>EAGLENA</ORGID>
        <SECTION />
        <STARTMEASUREUNITID />
      </ASSETSPEC>
      <ASSETSPEC>
        <ASSETATTRID>STAGE</ASSETATTRID>
        <CLASSSTRUCTUREID>1001</CLASSSTRUCTUREID>
        <LINEARASSETSPECID>0</LINEARASSETSPECID>
        <NUMVALUE xsi:nil="true" />
        <ORGID>EAGLENA</ORGID>
        <SECTION />
        <STARTMEASUREUNITID />
      </ASSETSPEC>
    </ASSET>
  </ASSET2Set>
</PublishASSET2>
```
Integration Framework – Using Integration Processing Rules to customize your integration processing.

```xml
<ASSETSPEC>
<ASSETSPEC>
<ALNVALUE />
<ASSETATTRID>SPEED</ASSETATTRID>
<classstructureid>1001</classstructureid>
<linearassetspecid>0</linearassetspecid>
<numvalue>1200.0</numvalue>
<orgid>EAGLENA</orgid>
<section />
<startmeasureunitid />
</ASSETSPEC>
<ASSETSPEC>
<ALNVALUE />
<ASSETATTRID>SIZE</ASSETATTRID>
<classstructureid>1001</classstructureid>
<linearassetspecid>0</linearassetspecid>
<numvalue>2.5</numvalue>
<orgid>EAGLENA</orgid>
<section />
<startmeasureunitid />
</ASSETSPEC>
<ASSETSPEC>
<ALNVALUE />
<ASSETATTRID>NPSH</ASSETATTRID>
<classstructureid>1001</classstructureid>
<linearassetspecid>0</linearassetspecid>
<numvalue xsi:nil="true" />
<orgid>EAGLENA</orgid>
<section />
<startmeasureunitid />
</ASSETSPEC>
<ASSETSPEC>
<ALNVALUE />
<ASSETATTRID>DRIVER</ASSETATTRID>
<classstructureid>1001</classstructureid>
<linearassetspecid>0</linearassetspecid>
<numvalue xsi:nil="true" />
<orgid>EAGLENA</orgid>
<section />
<startmeasureunitid />
</ASSETSPEC>
<ASSETSPEC>
<ALNVALUE />
<ASSETATTRID>CAPACITY</ASSETATTRID>
<classstructureid>1001</classstructureid>
<linearassetspecid>0</linearassetspecid>
<numvalue>100.0</numvalue>
<orgid>EAGLENA</orgid>
<section />
<startmeasureunitid />
</ASSETSPEC>
```
Integration Framework – Using Integration Processing Rules to customize your integration processing.

    </ASSETSPEC>
    </ASSET>
    </ASSET2Set>
    </PublishASSET2>

Asset 11450 was duplicated to create 11450B. This second asset contains the same assetspec values but has a different status and location:

<table>
<thead>
<tr>
<th>Assetnum</th>
<th>Status</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>11450</td>
<td>NOTREADY</td>
<td>BR450</td>
</tr>
<tr>
<td>11450B</td>
<td>OPERATING</td>
<td>PLANT-P1</td>
</tr>
</tbody>
</table>
Integration Framework – Using Integration Processing Rules to customize your integration processing.

**Processing Rule Definitions**

Within the Publish Channel application you can configure Processing Rules that will apply to messages flowing outbound using this Channel. A Processing Rule is created per object, based on the focused Object in the Object Structure Sub-Records section. There can be multiple rules per object as well as multiple rules for multiple objects. Processing rules are executed in the order of the object (starting at the root object) as they are configured in the channel’s object structure.

A processing rule can be configured with 1 of 8 possible Actions. Each Action provides specific functionality that will be described below. In addition a Channel can support having multiple processing rules with the same or different actions on each rule.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Other attributes of a Processing Rule include

<table>
<thead>
<tr>
<th>Rule</th>
<th>Unique Name of Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description</td>
</tr>
<tr>
<td>Sequence</td>
<td>Execution sequence of Rule (applies when multiple rules are configured)</td>
</tr>
<tr>
<td>Enabled</td>
<td>When Selected Rule is executed; when not selected Rule is ignored</td>
</tr>
<tr>
<td>Apply on Primary Object Insert*</td>
<td>Rule is applied only when the primary object of the object structure (Asset in our example) is being Inserted</td>
</tr>
<tr>
<td>Apply on Primary Object Update*</td>
<td>Rule is applied only when the primary object of the object structure (Asset in our example) is being Updated</td>
</tr>
<tr>
<td>Apply on Primary Object Delete*</td>
<td>Rule is applied only when the primary object of the object structure (Asset in our example) is being Deleted</td>
</tr>
<tr>
<td>Message Key</td>
<td>Key value of an existing error message – applicable only when the processing rule Action = STOP or SKIP</td>
</tr>
<tr>
<td>Message</td>
<td>Text of message associated to the Message Key</td>
</tr>
<tr>
<td>User Defined</td>
<td>Un-checked for product-supplied rules, checked for rules created as part of an implementation</td>
</tr>
</tbody>
</table>

* The ‘Apply on’ properties are only valid when the Publish Channel message is driven from an event (an insert/update/delete on the asset object). When the Publish Channel message is initiated from a Data Export, there is no insert/update/delete of the asset object and these settings are ignored and the processing will execute the processing rule.

The 8 values of the Action can be grouped into 2 buckets. The first is those that pertain to Conditions that are applied to Channel processing. For example, the channel processing will SKIP sending an asset message to an external system when the status is not equal to OPERATING. These conditions can be set by selecting Add/Modify Conditions button. The Action values that pertain to Conditions are: SKIP, SKIPRECORD, SKIPCHILDREN and STOP.

The second bucket of Actions are those that apply to setting values to the fields in the Channel message. These can be configured by selecting the Sub-Record Fields button. For example you can set field X in the Channel message by concatenating fields A and B separated by a delimiter value. The Action values that pertain to Setting fields are: SET, SPLIT, COMBINE and REPLACE.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

One Processing Rule can contain multiple Conditions, multiple Field Settings as well as a mix of conditions and field settings. For example, set field X by combining fields A and B separated by a delimiter, when the status of the asset equals OPERATING.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Assumption: Publish Channel is configured to be event-enabled.

Use Case: I want to send out asset messages to my external systems when an asset is created or updated but not when an asset is deleted.

The most basic of a rule for a SKIP would be to configure it without any conditions. You can configure a SKIP on the Asset object based solely on the ‘Apply on’ flags on the Rule. You can configure the rule to Skip and have the rule apply on Delete (not Insert or Update). What this would do is Skip event-based messages that were triggered by a delete of an Asset. Only Asset inserts and updates would not get skipped (asset message sent to the external system) since the rule would not be applied.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

SKIP Processing Rule using a Value

Use Case: I want to send out asset messages to my external systems when an asset status is set to OPERATING

Below is an example of a SKIP rule (configured to execute on Insert, Update and Delete) where the Channel message will be skipped (not sent) if the Status field in the XML is Not Equal to the value of OPERATING. This rule has 1 condition (10) where there is 1 evaluation (XML Field). Only assets that have a status of OPERATING will be sent to the external systems.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

SKIP Processing Rule using an Integration Control

Use Case: I want to send out asset messages to external system A when the asset status is set to OPERATING and send out asset messages to external system B when the asset status is set to NOT READY.

The processing rule in the prior Use Case used a ‘Value’ to compare to the Status field. By using a Value in the evaluation, all external systems using this Channel would have to live with this same condition.

To make this rule more configurable, you can replace the use of the Value with an Integration Control. By doing this, the control value for external system A could use the status value of OPERATING while external system B using this same channel could use the status value NOT READY.

Using the integration control allows multiple external systems to use the same Channel but get different processing behavior by supplying different values in the integration control, rather than having to create two Channels with different processing rules that have two different Values configured.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

The first screen shot below shows the ASSETSTATUS control being set to a value of OPERATING for ExtSysA and the second shows the same control for ExtSysB being set to a value of NOT READY. This configuration is done in the External Systems applications, which initially would have the control value that was defaulted when the control was created in the Publish Channel application.

For System A (ExtSysA), the value is set to OPERATING:
Integration Framework – Using Integration Processing Rules to customize your integration processing.

For System B (ExtSysB), the value is set to NOT READY:

So an asset message processing through this channel will get sent to ExtSysA if the asset asset’s status equals OPERATING, while ExtSysB will receive the message only if the asset’s status equals NOT READY.

Using a List Control instead of a Value Control
Since this integration control, ASSETSTATUS, is a ‘Value’ control it can only support 1 value. If you wanted to send data based on the asset status being set to one of multiple values, then a ‘List’ integration control could be used in the same manner as the Value control, as shown above. ExtSysA could have the values of OPERATING and CLOSED in its list while ExtSysB could have the values of NOT READY and CLOSED in its list. With this configuration, external system A would receive asset messages when the asset status is either OPERATING or CLOSED while external system B would receive asset messages when the asset status is either NOT READY or CLOSED.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Comparing Multiple Conditions versus Multiple Evaluations

Use Case: When does it make sense to use one condition with two evaluations versus using two conditions each with 1 evaluation?

The Add/Modify Conditions dialog below has 2 sections, Conditions (top) and Evaluations (bottom). The Conditions section allows you to create 1 or more conditions for a Processing Rule. Each condition can have 1 (minimum) or more evaluations. The result of the processing of the Conditions (1 or more) will return a True or False back to the Processing Rule. When a SKIP processing rule is returned a True from the Conditions, the Rule will skip the send of the message.

Multiple Conditions
When you have multiple conditions, they are evaluated as if there is an OR separating each condition. So if any one condition is satisfied, the result returned to the processing rule is True. For example you have a SKIP rule that has 2 conditions:

1. the first condition has 1 evaluation to skip when status is not equal to OPERATING
2. the second condition has 1 evaluation to skip when the Location equals BR450.

The results would be as follows:
Integration Framework – Using Integration Processing Rules to customize your integration processing.

<table>
<thead>
<tr>
<th>Location</th>
<th>Status</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR450</td>
<td>NOT READY</td>
<td>SKIP</td>
</tr>
<tr>
<td>BR450</td>
<td>OPERATING</td>
<td>SKIP</td>
</tr>
<tr>
<td>BR200</td>
<td>NOT READY</td>
<td>SKIP</td>
</tr>
<tr>
<td>BR200</td>
<td>OPERATING</td>
<td>Not Skipped</td>
</tr>
</tbody>
</table>

Multiple Evaluations
When you create 2 evaluations for 1 condition, the evaluations are treated as if there is an AND between the two evaluations. If we change the prior example to have a SKIP rule that now has 1 condition with 2 evaluations; the first evaluation is to skip when status is not equal to OPERATING and the second evaluation is to skip when the Location equals BR450. The results would be as follows:

<table>
<thead>
<tr>
<th>Location</th>
<th>Status</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR450</td>
<td>NOT READY</td>
<td>SKIP</td>
</tr>
<tr>
<td>BR450</td>
<td>OPERATING</td>
<td>Not Skipped</td>
</tr>
<tr>
<td>BR200</td>
<td>NOT READY</td>
<td>Not Skipped</td>
</tr>
<tr>
<td>BR200</td>
<td>OPERATING</td>
<td>Not Skipped</td>
</tr>
</tbody>
</table>
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Evaluations – Deep Dive

Evaluations for a Condition offer a number of configuration options that provide varying functionality. This section will cover the capabilities of Evaluations.

Evaluations are broken into four groups which are accessible on 4 tabs within Add/Modify Conditions dialog:

1. XML Field
2. Object Field
3. Object Set
4. Control

In the Conditions section of the UI (top), the four flags reference the four tabs in the evaluation section (bottom). When checked this identifies to the user that there is some evaluation data configured in that corresponding tab (informational, so the user doesn’t have to click each tab to see if configuration data already exists).
Integration Framework – Using Integration Processing Rules to customize your integration processing.

XML Field Tab
The evaluations for XML fields allows for the rule to set conditions against the field of the object XML that the processing rule is tied to. The field selected is from the XML message of the channel and not the corresponding field in the business object (mbo) from which the message was created from. In most cases, the value would be the same for both, however, the xml field value could be changed in the object structure definition class or by a processing rule that executes prior to the current rule being processed.

The fields available to select for an evaluation are only those that are ‘included’ in the object structure’s schema per the object structure definition.

This list of fields (17) shown from Field selection dialog shows only those Asset object fields that were configured for this object structure (sample XML shown earlier in this document).
Integration Framework – Using Integration Processing Rules to customize your integration processing.

When a Field is selected, its corresponding Data Type is shown as read-only. This data type is for the object field that the XML field is sourced from. The Evaluation Type defines the ‘operator’ that will be used with the selected field – there are multiple values to choose from:

![Integration Framework Screen](image)

The Evaluation Types listed above are commonly used in SQL-like conditions with the exception of NONE (this will be covered below).

The Field and Evaluation Type will be applied against one of the 5 choices selected on the right-side of the XML File Evaluation tab (Integration control, Value etc). The result of the evaluation will be either True or False.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

One additional configuration related to the Evaluation Type is the Evaluation When:

This allows the evaluation to be applied under 3 conditions:

1. ALWAYS
2. CHANGED – only when the selected Field had been updated
3. NOTCHANGED – only when the selected Field had not been updated.

For example, you may want to compare the Asset Status to the value in the ASSETSTATUS integration control only when the Status field has CHANGED.

CHANGED and NOTCHANGED only apply when the message is event-based (not via data export since there is no object update during an export). A field is considered Changed or Not Changed based on the update to that field in the business object (mbo) when the event was initiated. When Data Export is used, the processing rule is applied and the field is always considered Not Changed since the object is not changed during a Data Export.

The Evaluation type of NONE is available for the case where you only want to use CHANGED or NOTCHANGED. When you select NONE, the options on the right side to compare the selected Field to are not select-able. Using NONE means you are only evaluating whether the field has been updated or not updated; you are not comparing the field value to anything else. For example, you may want to send the asset channel message only when the Status field has CHANGED, regardless of what value the status field contains.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

On the right side of the XML Field tab there are 5 choices to use to apply the field and Operator against.

1. Integration Control – As shown in an earlier example, an XML field can be compared to a value stored in an integration control. The look-up for the integration control will allow you to select any Value or List control, however, you need to remember to associate the control to the channel which pushed the default value(s) to all external systems using that channel.
2. Value – the XML field can be compared to the value typed into this field
3. MAXVAR – the XML field can be compared to the value that is held in a system maxvar (the MAXVAR table).
4. Comparison Field – the XML field can be compared to another XML field of that object in the XML message. The list of fields available to compare against is limited by object structure definition (only those fields that are included in the object structure).
5. Object Field – the XML field can be compared to any field in the same object or a different object when that other object can be obtained using an existing object relationship. When comparing to a field in the same object, no relationship name is needed. All fields of the selected object are available to compare against.

If the XML field you’ve selected (configured on the left side) is a Numeric data type and the value it is compared to (configured on the right side) is a non-numeric you
Integration Framework – Using Integration Processing Rules to customize your integration processing.

will receive a message like this (this example is a number field being compared to a value of ‘abc’):

You will get a different (but similar) message when you compare a Date field to a value that is not of a Date format.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

Use Case: I want to send out an asset message any time the status of an asset changes (I don’t care what the value of the status is, I just want to notify the external system that the asset’s status has changed).

The screen shot below shows a condition of a SKIP processing rule where the STATUS field is being evaluated for not being changed (NOTCHANGED). When true (status not updated), the processing will skip the sending of the message. When false (status did get updated), the processing will not skip the sending of the message.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

SKIP Processing Rule comparing an XML field to a MBO field of a different business object.

Use Case: I want to send out an asset message when the asset is updated but only if the Status of the asset is equal to the Status of its location.

The screen shot below shows a condition of a SKIP processing rule where the STATUS field (of the asset which the rule is configured for) is being compared to the STATUS field of the asset’s location. Using the Object field to identify the LOCATIONS object and selecting a valid relationship (maxrelationship) between the asset and locations objects allows the rule to obtain the status field of the location for comparison. When true (status fields not equal), the processing will skip the sending of the message. When false (status fields equal), the processing will send (not skip) of the message.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

SKIP Processing Rule comparing a related MBO field of a different business object to a Value

Use Case: I want to send out an asset message when the asset is updated but only if the Asset’s Location has a status of OPERATING.

The screen shot below shows a condition of a SKIP processing rule (on the Asset MBO) where the STATUS field of the Asset’s Location MBO is being compared to the value of OPERATING. This capability allows access to an object field even though that object is not included as part of the object structure of the channel. Using the Object field to identify the LOCATIONS object and selecting a valid relationship (maxrelationship) between the asset and locations objects, allows the rule to obtain the status field of the location for comparison. When true (status not equal OPERATING), the processing will skip the sending of the message. When false (status equals OPERATING), the processing will send (not skip) of the message.

The configuration allows the choice using an integration control or a Maxvar to be specified in addition to a Value being provided.
Integration Framework – Using Integration Processing Rules to customize your integration processing.

**SKIP Processing Rule based on the existence of a related MBO**

Use Case: I want to send out an asset message when the asset is updated but only if the asset is configured with 1 (or more) active meters.

The screen shot below shows a condition of a SKIP processing rule (on the Asset MBO) where the condition is checking to see if there is 1 (or more) occurrences of the ASSETMETER that relate to the asset being updated. The selected relationship includes criteria to select only meters that are configured as Active. Using the Object field to identify the ASSETMETER object and selecting a valid relationship (maxrelationship) between the asset and assetmeter objects, allows the rule to determine if at least 1 meter exists for the asset being updated. Selecting NOTEXISTS as the Evaluation Type directs the rule to SKIP the sending of the message when no active meters exists for the asset. If one (or more) active meters exists for the asset, the processing would send (not skip) the asset message.

The configuration also allows the selection of EXISTS as the evaluation type in addition to NOTEXISTS.