

*IdentityIQ Accelerator Pack Ticket Integration*

Table of Contents

1. Overview 3

2. After Ticket Creation Rule 3

3. Assignment Group Name 3

4. Simulated Ticket Integration Executor 3

5. Plan Initializer Rule 4

6. Ticket Integration 5

7. Ticket Maximum Status Checks 6

8. Check Ticket Status 7

9. Global Ticket Configuration 7

**Document Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Author** | **Summary of Changes** |
| Version 1 | 3/2/18 |  | Rohit Gupta |  |
| Version 2 | 4/4/18 |  | Cathy Mallet | Minor grammar/spelling/etc edit |

## Overview

The ticket integration feature built within the Accelerator Pack supports the following functionalities:

* After Ticket Creation Rule Per Application
  + This is not applied to certification
* Global Ticket Properties via Global Services Definition
  + Assignment Group Name
  + Ticket Polling Maximum Retries
  + Ticket Status Check Interval in Minutes
* Application Ticket Properties via the Self Service Onboarding Wizard
  + Assignment Group Name
  + Ticket Polling Maximum Retries
  + Ticket Status Check Interval in Minutes
  + Default Assignment Group Name and Requestor
* Check Ticket Status QuickLink
* Ticket Integration QuickLink
  + User Interface
  + Plan Initializer Rule
* Simulated Ticket Integration Executor

## After Ticket Creation Rule

This rule can be executed per application. It is defined via the Application Administrative Wizard. The internal extended attribute name of the application that is used for the rule reference is “afterticketprovisioningExtendedrule”. This rule is executed after ticket creation and it will only be executed when application is configured for the “Extended Rule” option. The internal extended attribute name of the application that is used for the option reference is “afterTicketProvisioningOptions”.

## Assignment Group Name

Every application requires dedicated work queues or assignment group names on ticketing systems. This is configured per application via the Self Service Onboarding Wizard. The internal extended attribute name of the application that is used for the assignment group name reference is “assignmentGroupname”.

## Simulated Ticket Integration Executor

This executor is used for testing the ticket integration configuration for applications within IdentityIQ. It has two components

* IntegrationConfig: SmartServicesTicketIntegration
* Java Executor: sailpoint.rapidapponboarding.testing.ROADTicketIntegrationExecutor

This simulator creates a ticket number “123456789” for all application provisioning requests and uses rule “Rule-Framework-Plan-Initializer” to insert data into the provisioning plan as arguments.

## Plan Initializer Rule

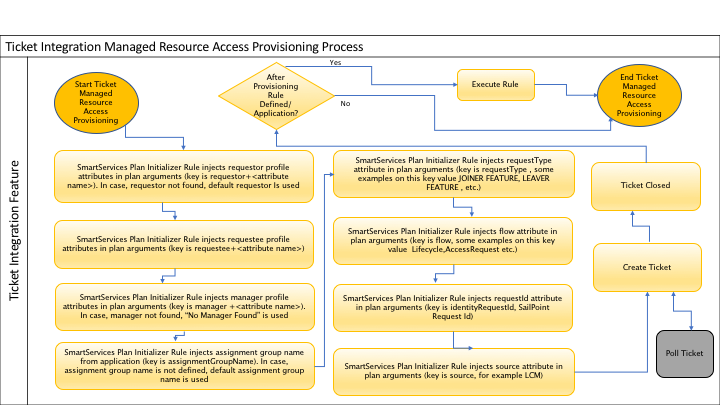
This rule is used to insert request data into a provisioning plan as arguments. It is a requirement to configure this rule as a initializer rule for all Ticket Integrations.

It uses a default requestor configured using [Ticket Integration](#Ticket Integration) if the request is coming from a background event.

It uses the default assignment group name if it is not configured on the application via the Self Service Onboarding Wizard.

The SOAP envelope configuration is simplified using this rule. The SOAP Envelope for the “provisioning” entry and the “getRequestStatus” entry in the integration configuration can be configured with the following velocity parameters:

* Requestor Identity Attributes
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘requester<attributename>’)}
* Requestee Identity Attributes
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘requestor<attributename>’)}
* Requestee’s Manager’s Identity Attributes
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘manager<attributename>’)}
* Assignment Group Name for the requested Application
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘assignmentGroupName’)}
* IdentityIQ Request Id
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘identityRequestId’)}
* “requestType” variable initialized from the workflows
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘requestType’)}
* “flow” variable initialized from the workflows
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘flow’)}
* “source” variable initialized from the workflows
  + Referenced in SOAP Envelope as $!{provisioningPlan.getArguments().get(‘source’)}



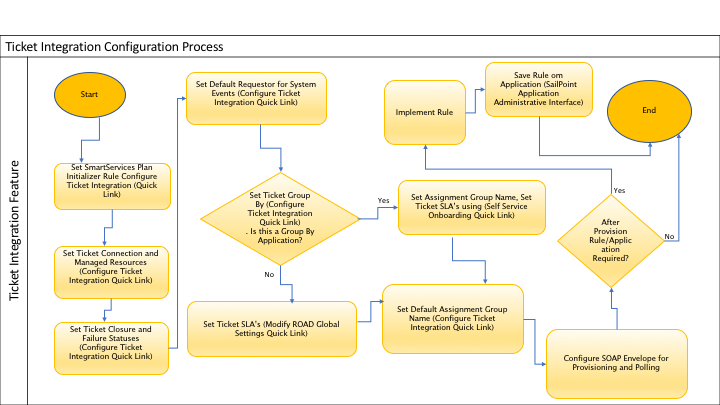
## Ticket Integration

A new QuickLink is designed for configuring “IntegrationConfig” via a QuickLink form.

The ”Ticket Integration” QuickLink provides a user interface to set up connection settings for ticket integration and to set up applications for ticket integration. The form fields of this user interface are configured using the custom artifact “Custom-SmartServices -TicketIntegration-Form-Settings”. The configuration is based on key parameters for each type of integration. These settings are stored in the custom object “**Custom-SmartServices-TicketIntegration-Form-Settings**” artifact. This custom artifact can be configured from the Debug page if there is any upgrade/patch change to the integrations.

* Common Executor – the Key Parameter in the Custom Artifact is “commonExecutor”. This key defines common fields that will be displayed for all integrations
* Remedy – the Key Parameter in the Custom Artifact is “RemedyIntegrationExecutor”. This key defines only Remedy Ticket Integration fields that will be displayed on the form
* ServiceNow - the Key Parameter in the Custom Artifact is “ServiceNowIntegrationExecutor”. This key defines only ServiceNow Ticket Integration fields that will be displayed on the form
* HP Service Manger- the Key Parameter in the Custom Artifact is “HPServiceManagerIntegrationExecutor”. This key defines only the HP Service Manager Ticket Integration fields that will be displayed on the form
* Accelerator Pack Simulated Ticket Integration (Used for Testing)– the Key Parameter in the Custom Artifact is “ROADTicketIntegrationExecutor”. This key defines only Accelerator Pack Simulated Ticket Integration fields that will be displayed on the form

|  |  |  |
| --- | --- | --- |
| Property Name | Description | Default Settings |
| ServiceNowIntegrationExecutor | This key has comma separated QuickLink form field names for ServiceNow Ticket Integration | keystorePath,keystorePass,keystoreType,alias,keyPass,catalogItem,catalogItemList,username,password,authType |
| HPServiceManagerIntegrationExecutor | This key has comma separated QuickLink form field names for HP Service Manager Ticket Integration | multipleProvisioningSteps,lastProvisioningStep, checkStatusProvisioningStep,catalogItem,catalogItemList,username,password,committedClosureCode,failureClosureCode,inProcessClosureCode |
| RemedyIntegrationExecutor | This key has comma separated QuickLink form field names for Remedy Ticket Integration | basicAuthType,httpUserName,httpUserPass |
| ROADTicketIntegrationExecutor | This key has comma separated QuickLink form field names for Accelerator Pack Simulated Testing Ticket Integration | Empty |
| commonExecutor | This key has comma separated QuickLink form field names for all Ticket Integration Systems | executor,universalManager,planInitializerRule,multipleTicket,groupTicketBy,retryableErrors,noProvisioningRequests,provisioningRequestExpiration,managedResources,operations,defaultAssignmentGroupName,defaultRequestedBy,inProcess,failure,committed,ticketStatus,ticketId |



## Ticket Maximum Status Checks

An error message “**Retry Timeout**” is sent out to Operations team in case ticket is not acted timely. This is based on “Ticket Status Check Interval in Minutes” either configured globally or per application.

A new step “Set Timeout Error” is added to the subprocess “Interactive Request Check Status of queued items” to add error message to the workflow arguments.

## Check Ticket Status

A new QuickLink named “Check Ticket Status” is designed to check ticket status based on a selected IntegrationConfig object.

The workflow “Workflow-QuickLink-Ticket-Status” uses the Java Class “CheckTicketStatusExecutor” to calculate ticket status. The ticket status is calculated based on the following selected ticket IntegrationConfig Executors:

* ServiceNowIntegrationExecutor
* HPServiceManagerIntegrationExecutor
* RemedyIntegrationExecutor
* ROADTicketIntegrationExecutor

## Global Ticket Configuration

The application ticket settings for Ticket Polling Maximum Retries and Ticket Status Check Interval in Minutes takes **precedence** over global ticket settings configured via “Global Services Definition” Quick Link.

**Please Note:** Provisioning Feature “Provisioning Maximum Retries” and “Provisioning Retry Interval in Minutes” is applied to Ticket Creation Provisioning