System Administration

Version: 8.2

Revised: June 2021

Copyright and Trademark NoticesCopyright © 2020 SailPoint Technologies, Inc. All Rights Reserved.

Contents

Using the Administrator Console **4**

Manage Task Results 4

Active Tab 4

Scheduled Tab 5

Complete Tab 5

Manage Provisioning Transaction Results 6

Monitoring Your Environment 7

Hosts 7

Specifying Hosts to Handle Requests 8

Applications 8

SailPoint Modules and Extensions 9

Partitioning **10**

Loss Limits 10

Configuring Partitioning Request Objects 11

Alerts **12**

Alerts Page 12

Create Alert Definition 13

Procedure 14

Edit Alert Definitions 14

How to Edit an Alert Definition 15

Procedure 15

How to Filter Alerts 15

Procedure 15

Using the Administrator Console

Use the Administrator Console link, under the gear icon, to access the Administrator Console and view the Task, Provisioning, and Environment monitoring tables.

- Manage Task Results
- Manage Provisioning Transaction Results
- Monitoring Your Environment

Access to the Administrator Console is controlled with IdentityIQ rights.

Manage Task Results

Use the Tasks table to view host affinity check run time data. From this page you can also postpone a scheduled task, terminate a running task, or dump a stack trace of a running task. The stack trace is typically used when a task is running long and diagnostics are needed.

Use the tabs at the top of the table to limit your view by task status: Active Tab, Scheduled Tab, or Complete Tab. Use the **Filter** options or search field to further limit the tasks displayed.

Active Tab

This tab displays all of the tasks that are currently running.

Use the Actions column to terminate a running task or request a stack trace, if a task is running long and you would like to see diagnostics.

Name

Name of the task

Type

Task type

Start Date

Name of the task

Owner

The task owner, not necessarily the identity who requested the task be run

Host

Host on which the task is currently running

Current Runtime

How long the task has been actively running

Average Runtime

The time that this task has historically taken to complete

Scheduled Tab

This tab displays all of the tasks that are scheduled to run in the future, including those that are scheduled to run periodically, for example Perform Maintenance.

Use the Actions column to postpone a scheduled task or delete a schedule. No instance of a postponed task will be performed until after the selected date.

Name

Name of the task

Type

Task type

Task

Name of the task

Host

Host on which the task is scheduled to run

Next Execution

The next time this task is scheduled to run

Last Execution

The last time this task was executed

Last Result

The result of the last run, for example Success or Failed

Owner

The task owner, not necessarily the identity who requested the task be run

Complete Tab

This tab displays all of the tasks that have completed, regardless of the result.

Name

Name of the task

Type

Task type

Result

The result of the last test run

Start Date

The date and time at which this task began

Date Complete

The date and time at which this task stop running

Owner

The task owner, not necessarily the identity who requested the task be run

Host

Host on which the task was run

Average Runtime

The time that this task has historically taken to complete

Runtime

The actual runtime

Diff from Average

The difference between the actual and average runtimes

Manage Provisioning Transaction Results

> This feature can be disabled and might not appear in your instance of IdentityIQ. Contact your system administrator for details.

Use the Provisioning Transactions table to view the status of all provisioning transactions in your implementation of IdentityIQ; connectors, manual work items, and IdentityIQ operations.

Use the tabs at the top of the table to limit your view by transaction status: All, Failure, Success, or Pending. Use the **Filter** options or search field to further limit the transactions displayed. The logging level is controlled by a system setting. If you are not seeing all of your transactions, contact your system administrator.

Use the report/download button to launch a Provisioning Transaction Object report in the background. From the Report Launched window, use **Get Email Notification** to receive an email when the report is complete, or **View Report** to display the Report Results page.

Click the information icon for any transaction to view detailed information. The Transaction Details window provides very detailed information, including the reasons for a Failed or Pending status. After viewing take the appropriate actions to correct the reasons for the failure or delay, you can use the **Override** or **Retry** options to proceed with the provisioning process.

Use **Override** to manually create a work item to take action on failed transactions.

Use **Retry** to manually retry the provisioning transaction. The retry option is only available on transactions that are in the Pending state, and only for transactions that were created with the retry options enabled. This button overrides the reset counter configured in the transaction.

Failed transaction cannot be retried, you must use the override option to create a new work item for those transactions.

The information contained on this page is also available in two reports, the Provisioning Transaction Object Report and the Detailed Provisioning Object Report.

Manage the information displayed on this page from the Miscellaneous tab of the Configure IdentityIQ Settings page of Global Setting, found under the gear icon.

Monitoring Your Environment

The Environment Monitoring page provides insight into each defined Application's health and the status of your Modules and Extensions. This helps diagnose issues with connectivity within the environment.

The Application view provides a view from an Application up perspective. Each Application is listed, along with a summary of all statuses reported by all configured Hosts.

Click **Columns** to select and arrange the information displayed on the pages. Use the search field to locate specific information.

Use the gear icon in the title bar to define global settings for all hosts in IdentityIQ. These settings are used for all hosts that have not explicitly over-ridden the defaults.

Hosts

Specifying Hosts to Handle Requests

Applications

SailPoint Modules and Extensions

Hosts

This tab displays all of the hosts associated with an IdentityIQ instance.

Click on a name in the **Host Name** column to show all ServerStatistics captured for the selected host, grouped by Snapshots. The snapshots can be cycled using the previous/next arrows, or selected by name using the drop-down list.

Use the action buttons in the **Host Action** column to configure or delete hosts. The Host Setting dialog enables you to specify the services running on each host and configure host monitoring.

Deleting a host will remove all associated server statistics, as well as the Server object. The host will no longer appear in the list of hosts after deletion. However, if the underlying server is still running, the host will reappear the nest time its heartbeat service runs. All configuration settings for a re-generated host will use defaults for the its list of services, and for the monitoring service configuration.

Services

This enables a specific host to enable/disable services. The one exception is the Request Service. The Request Service cannot be fully shut down. The Host, if service shutdown, still processes requests that have been specifically targeted to that host, but will not pick up generic requests (un-targeted requests).

Reanimator Sevice for Tasks

The Reanimator service helps manage “hung” tasks. An un-partitioned task can sometimes fail without properly updating the state of the TaskResult. This can leave the task in a “hung” state, appearing to still be running even though it isn’t. The most common cause for this kind of issue is a temporary loss of connection to the database, or a brief database server failure. However, the Reanimator service performs the task of resetting requests or tasks regardless of the underlying reason.

When a task has hung, the Reanimator service resets the task or request so that it can resume. The service can also help with the termination of a task or request that is not configured to resume upon being orphaned or failing. If the task or request is not configured to resume, then when the service detects a task in a hung state, it automatically marks it as terminated.

The Reanimator service runs by default on all hosts. Although it is unlikely that you would need to switch it off, it is possible to do so – for example, if you have a dedicated UI host, you might not need this service running there. To disable the Reanimator service on a specific host:

1. On the **Hosts** tab, click the **gear** icon beside the specific host on which you want to disable the service.
2. On the **Services** tab of the Host Configuration dialog, use the slider to switch off the Reanimator service.

Configuration

> The Application Monitoring does not adhere to the restore defaults.

The Configuration tab enables host specific monitoring configuration. This enables you to override the global defaults for Polling Interval and Statistics Retention, and to enable and disable given retained statistics.

Click **Use Default Settings** to clear all host specific overrides revert back to the global defaults.

The Configuration tab also allows selecting Applications in which to monitor health. When selected, the Application is contacted each time the monitoring service runs, and the health check status is recorded.

Specifying Hosts to Handle Requests

In a multi-host environment, you can specify which hosts can process specific types of requests, by including a list of hosts (or a single host) in a RequestDefinition object. This can help with performance, allowing you to dedicate specific machines and threads for processing request types that are, for example, operating at a high volume or require more resources.

1. In the Debug pages, choose RequestDefinition in the **Select an Object** field of the Object Browser.
2. From the list of RequestDefinition objects, choose the object that you want to modify in order to specify a host or list of hosts.
3. Add host entry to the RequestDefinition attribute map. The value can be set to a single host, or multiple hosts separated by commas. For example:

```
<Attributes>
 <Map>
 <entry key="hosts" value="hostA,hostB,hostC"/>
 <entry key="maxThreads" value="1"/>
 </Map>
 </Attributes/>
```

4. Save the RequestDefinition object.

Applications

> An application must be monitored by at least one host before it will report statistics.

The Application tab provides a view from an application up perspective. Each application is listed, along with a summary of all statuses reported by all configured hosts. Monitoring can be run from any number of servers, on any subset of applications.

Click an application name to display a panel containing more detailed information about a the application's statuses. This shows each host that has reported a status for the application, as well as the status, and time of ping.

If you have full access rights, click the refresh icon to schedule a request on the host to perform a health check for the application. The refresh icon is disabled until the request is fulfilled.

SailPoint Modules and Extensions

The SailPointModules and Extensions tab provides a list of all installed modules and extensions and a summary of all statuses reported. Click a module or extension name to see a list of reported statuses.

Partitioning

> Partitioning is not available on all task or certifications. Partitioning is available for Account Aggregation, Account Group Aggregation, Identity Refresh, Perform Identity Request Maintenance, and Manager Certification generation.

> Partitioning is not available on all application types. Partitioning is controlled by both the configuration of the applications you are using and the configuration of the connectors used to communicate with those applications.

Partitioning is used to break operations into multiple pieces, or partitions. Each partition is then placed in a global queue, and machines, or hosts, in a cluster compete to execute the partitions in the queue. Machines are added or removed from the cluster dynamically with automatic balancing. If a machine fails or is taken down while processing a partition, the partition is placed back into the queue and reassigned to a different machine. A single result object is shared by all partitions and is continually updated so you can monitor the overall progress of the partitioned operation. When all partitions have finished executing the result is marked complete.

Each instance of IdentityIQ includes a Server object containing information about what is happening in that instance. For machines running multiple instances of IdentityIQ, you must give each instance must be assigned a unique iiq.hostname and have a unique Server object.

The Server objects include a heartbeat service and is updated by a new system thread on a regular basis. By monitoring server heartbeats, machines in the cluster can detect when another machine fails. When this happens any partitioned requests that were running on that machine are restarted and picked up by a different machine in the cluster, so that failure of one machine does not terminate an entire long running task.

Server objects include some statistics, such as the number of request threads currently active and the request types that are executing. You can view the state of the machines in your cluster on the Administrator Console page. See Using the Administrator Console.

To activate partitioning you must have applications configured for partitioning, connectors configured to work with those applications, and you must enable partitioning when defining an account aggregation or identity refresh task, or scheduling a manager certification.

- Applications are configured as part of the Account Settings on the Configuration tab of the Application Configuration page. See the **Application Configuration** documentation.
- For task details, see the **Tasks** documentation.
- For certification details, see the **Certifications** documentation.

Loss Limits

Configuring Partitioning Request Objects

Loss Limits

Some of the features which support Partitioning also include an option to set loss limits for the identities or accounts being processed by a task. The loss limit sets the maximum number of identities or accounts that will be reprocessed in case of a sudden termination of a partitioned refresh.

In a partitioned task, each time the task reaches the loss limit – that is, it has processed a number of accounts or identities that match the value of the loss limit – it commits a list of the accounts to a `requestState` object. If the task

should happen to fail, due perhaps to a server or database going down, the task will check the `requestState` object when it resumes, so that it knows which accounts have already been processed. This means the task doesn't have to re-process the entire partition. A lower loss limit number will result in less duplicated work following a crash, but may slow down the task due to increased database contention.

Loss limit data that is stored in the `requestState` object is base-64 encoded and so is not human-readable. RequestState objects are not retained in the IdentityIQ database past their usefulness; in other words, once a loss limit has been reached, the object for that particular segment is automatically deleted.

Configuring Partitioning Request Objects

Partitioning is also maintained using RequestDefinition objects that are defined for each request type. These objects control how each request-type is processed. For example, these objects define the number of threads that run for each request on the instances of IdentityIQ running on a specific machine. The RequestDefinition objects must be defined on each machine, host, in a cluster.

> By default the maximum number of threads to run on each host is set to 1. This number can be changed to maximize performance in your environment, but should be done with caution and only after testing and tuning for your environment.

The following RequestDefinition objects are available:

- Aggregation Partition— define the maximum number of threads to run on each host during account aggregations
- Identity Refresh Partition — define the maximum number of threads to run on each host during identity refresh
- Manager Certification Generation Partition — define the maximum number of threads, the error action, and orphan action for partitioned manager certification requests
- Role Propagation Partition — define the maximum number of threads to run on each host during role propagation

To work with the RequestDefinition objects, go to the IdentityIQ Debug page and select RequestDefinition from the **Select an Object** drop-down list.

Alerts

Alerts are created using IdentityIQ File Access Manager (FAM) based on activity data - actions users take on resources that are part of an application that FAM is monitoring. FAM can be configured to create alerts when the user action is considered unexpected, potentially risky, or inappropriate. It is possible to configure alerts for any behavior. You can choose to use this functionality more broadly (e.g. for non-risky or non-problematic activities that someone wants to use as a process trigger).

This integration additionally enables you to trigger actions in IdentityIQ in response to an alert. Specifically, alerts aggregated into IdentityIQ can be used to drive three different response actions. A single alert can trigger more than one response action:

- Launch a certification
- Launch a workflow
- Send an email notification

Alerts Page

Alert Definitions Page

Edit Alert Definitions

Alerts Page

Use the Alerts page to view existing alerts for your enterprise. To limit the number of alerts displayed in the table, use the filtering options.

Alert Page Details:

Name

The name of the alert.

Source

Application associated with the alert.

Native Id

Native identifier of the application with which the alert is associated.

Type

Alert type.

Target Type

Type of the object that triggered the alert.

Target Name

Name of the object that triggered the alert.

Alert Date Start

Date and time at which this alert was triggered.

Alert Date End

Date and time at which this alert expires.

Last Processed Start

Last date and time this alert was triggered.

Last Processed End

Last date and time this alert process finished.

Acted Upon

Select True if this alert matched an alert definition and an alert action was triggered.

Create Alert Definition

The Create Alert Definition page contains the following information:

Details:

Name

A descriptive name of this alert. This is the name that displays on the Alerts page.

Display Name

Label that is displayed on the alert.

Description

A brief description of the alert.

Owner

The alert owner, not necessarily the identity who triggered the alert.

Match Rule

Enables more complex matching logic.

+Add

Option to add a **Match Term**.

Source

Application name that triggers the alert.

Attribute

The display name of an account attribute derived from the attribute and its associated application.

Value

Value for the selected attribute that will trigger an alert during alert processing.

Action Type

Action to be taken when the alert is created. This can either be a notification, certification, or a workflow, or a combination of the available actions.

Email Template

Template used for the notification email. If none is selected, a system default is used.

Email Recipients

List of users to receive the alert notification.

How to Create an Alert Definition

Alerts are created using the Alert Definitions tab. Use this procedure to create new alert.

Procedure

1. Click the **Alert Definitions** tab on the Alerts page.
2. Click **+New**.
3. Enter the alert information.
4. Click **Save** to save the alert and return to the Alerts page.

Edit Alert Definitions

Use the Edit Alert Definition to edit existing rules. The Edit Alert Definitions page contains the following information:

Name

A descriptive name of this alert. This is the name that displays on the Alerts page.

Display Name

Label that is displayed on the alert.

Description

A brief description of the alert.

Owner

The alert owner, not necessarily the identity who triggered the alert.

Source

Application name that triggers the alert.

Attribute

The display name of an account attribute derived from the attribute and its associated application.

Value

Value for the selected attribute that will trigger an alert during alert processing.

Match Rule

Enables more complex matching logic.

Action Type

Action to be taken when the alert is created. This can either be a notification, certification, or a workflow, or a combination of the available actions.

Workflow

Defines the workflow structure and steps involved in the workflow processing.

Email Template

Template used for the notification email. If none is selected, a system default is used.

Email Recipients

List of users to receive the alert notification.

How to Edit an Alert Definition

How to Filter Alerts

How to Edit an Alert Definition

Alerts are edited using the Alert Definitions tab. Use this procedure to edit existing alerts.

Procedure

1. Click the **Alert Definitions** tab on the Alerts page.
2. Select an alert and lick **Edit** in the Actions column.
3. Enter the alert information.
4. Click **Save** to save the alert and return to the Alerts page.

How to Filter Alerts

Use the filtering options to limit the number of alerts displayed in the table. You can filter by any field. Use this procedure to filter through existing alerts.

Procedure

1. Click the **Alert** tab on the Alerts page.
2. Click **Filter**.
3. Enter filtering information.
4. Click **Apply** to save the filter options.