Modify your Observer actions in ServiceNow



The **Observer Actions** are a way to track any number of metrics on your ServiceNow instance. These metrics are then packed into messages and sent through MBS to access through your Observer portal.

Available Actions

Tracking Tasks

| Action | Description | Purpose | Interval |
|--------------------------------------------|-----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|----------|
| Active Requests with Closed Items | Will poll the request table for those which have requested items which are no longer active. | Keep track of requests which are no longer valid/necessary. | Hourly |
| Caller is deactivated on incidents | Will poll the incident table for active records whose caller is no longer active | To track the count of incidents where the caller is no longer valid | Hourly |
| Approval on Closed Tasks | Will poll the Approval table for approval requests on an inactive task | Keep track of approvals which were not completed in time or properly. | Hourly |
| Approver is deactivated | Will poll the Approval table for approval requests on where the assigned approver is inactive | Keep track of approvals which were not assigned properly. | Hourly |
| Open Tasks with Assigned to Deactivated | Will poll the Task table for those which are assigned to an inactive user | To track when tasks are improperly assigned. Can also be easily modified to track the count of unassigned tasks. | Hourly |

Users and Access

| Action | Description | Purpose | Interval |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|----------------|
| Local User Accounts | Will poll the User table (sys_user) for a count of local users, by keying off the "source" column. | Track the growth of local users on an instance. | Daily |
| External User Accounts | Will poll the User table (sys_user) for a count of external users, by keying off the "source" column. | Track the growth of external users on an instance. | Daily |
| Inactive Users | Will poll the User table (sys_user) for a count of user's who have not logged in the last 60 days | Track user access/retention on an instance. | Daily |
| Aggregate Active User Roles | Will aggregate the data for the count of user's who have been granted each role (sys_user_has_role) | To track how much access has been granted to certain areas of the instance. | Daily |
| Application Acc ess Count | Will aggregate the data for the count of user's who have accessed certain applications, per application. Done through polling the app usage table (ua_app_usage) | To track the traffic of each application | On Interval |

Activity and Response Metrics

| Action | Description | Purpose | Interval |
|----------------------------|------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|----------------|
| Import Set Queue | Over the last 60 seconds how many import sets (sys_import_set) are in each state (processed/loading /loaded) | Track how "backed up" the instance is in processing its import sets. | On Interval |
| Import Set Run Queue | Over the last 60 seconds track the counts of all the states of the current Import Set Runs (sys_import_set_runs) | Track the current processing of the Import Sets. | On Interval |
| Customer Updates | Over the last 60 seconds how many customer updates were made (those which are tracked in sys_update_xml) | Track how much work is being performed on the instance. | On Interval |
| Email Queue | Over the last 60 seconds track how many emails (sys_email) are in each state are in the instance. | Track how many emails you are generating compared to how quickly you are sending them out. | On Interval |

| Observe Nodes | Track the computing/sql metrics of each node (as well as averaged) on the instance | Track the CPU usage, response metrics, SQL response metrics, available semaphores, worker threads, etc. Type "stats.do" in the filter navigator for an example. | On Interval |
|------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
|------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|

Other

| Action | Description | Purpose |
|------------------------|----------------------------------------------|---------------------------------------------------------------------------------|
| Run on Login Failed | Ran when the event "login.failure" is fired | Track login failures on an instance as well as the common users which see this. |
| Daily Health Check | Ran nightly to capture a common set of data. | You can read more about the metrics here. |

Procedure

To make changes to the **Observer actions**, follow these steps:



Navigate to Perspectium Observer Properties

Log into your ServiceNow instance and navigate to Perspectium > Observer > Actions.



Select Observer action

Click the name of the Observer action you want to modify.



Modify the script (Optional)

To control when to send out metrics and how to send out metrics, you can add conditions to where metrics are generated to be shared out from your ServiceNow instance to the Perspectium Integration Mesh.

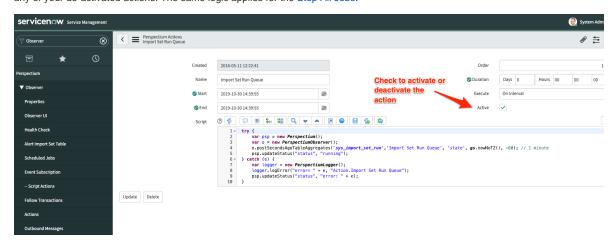
In the above example, conditions are added to calling **o.postTuple()** so that Application Access Count metrics are only sent out if the metric is for an application/table (the **app_name** field of a ua_app_usage record) named **u_psp_observer_out_message** and a count greater than 5 or if it's for **perspectium_properties** and the count is greater than 10. For questions configuring conditions, please contact Perspectium Support.

4

Enable or disable the action

Actions will only execute if their active flag is set to true. You may de-activate the metrics which you do not care about without impacting any of the other metrics.

(i) NOTE: Start All Jobs will re-activate the scheduled jobs responsible for processing these actions, but it will NOT re-activate any of your de-activated actions. The same logic applies for the Stop All Jobs.



Similar topics

- Create a ServiceNow bulk/dynamic share for Splunk
 Open Splunk HTTP Event Collector port to the Perspectium Integration Mesh
 Generate a Splunk Event Collector token
 View your event collections in Splunk
 Create custom Freshservice ticket fields

Contact Perspectium Support



US: 1 888 620 8880

UK: 44 208 068 5953

 ${\bf support@perspectium.com}$