

Application version: 1.1

Note: for version 1.1, there is also a YouTube video that walks you through the installation.

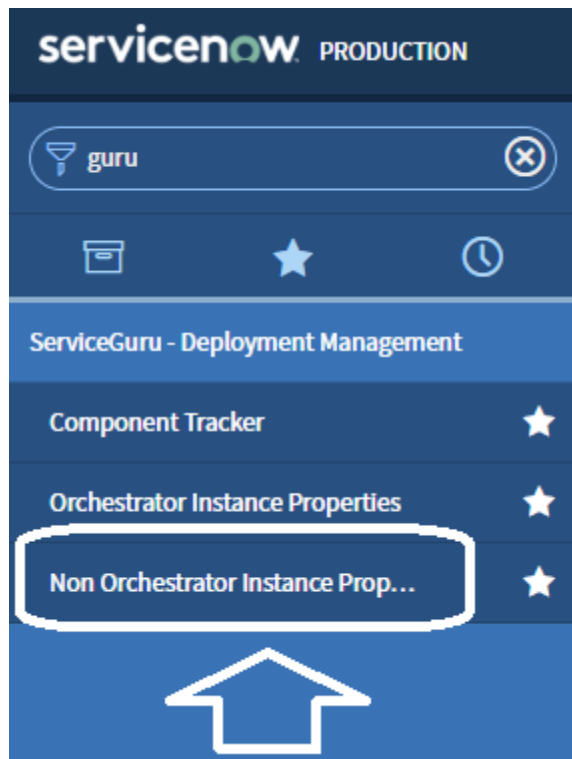
Link: <https://youtu.be/MQENNI2y2Bs>

Prerequisites:

- 1) System property 'glide.update_set.auto_preview' must be set to true in all environments which are in scope for the integration.
- 2) Install an integration user account in all your instances. This account will be used for retrieving the update sets, committing update sets and committing scripts. The admin role would be ideal but check to see if a lower role is sufficient. Ensure that the username and password are the same across all instances. If you need to provide a unique username and password combination for different environments, refer to the FAQ document.
- 3) Create Update sources for all environments from where you expect to retrieve the update set. If we have the code movement path of Dev to QA, QA to UAT and UAT to Prod. We need to have update sources in QA, UAT and Prod for the previous instances.

Non-Orchestrator instance installation steps

- 1) Install the batch update sets (update set name: ServiceGuru-Shell) in all the instances which are in scope for the integration.
- 2) Download the second update set 'ServiceGuru-FIRST-TIME-INSTALL-ONLY' from [here](#). Upload and install this update set. This is a non-batched single update set and contains jobs and properties for the first time installation. For subsequent upgrades of this application, you don't need to install this update set.
- 3) Type 'ServiceGuru' in the left navigation and click on 'Non-Orchestrator Instance Properties' properties.



a. You should see the configuration page like the one below

servicenow PRODUCTION

ServiceGuru

Auto Deploy

RS Ravish Shetty

Search

Help

Settings

guru

Component Tracker

Orchestrator Instance Properties

Non Orchestrator Instance Prop...

ServiceGuru - Non-Orchestrator Properties

Save

Important: The changes to properties on this page are local to this instance. You will need to perform the same property changes in respective ServiceNow instances where this tool is installed.

Contact: shettyravish@outlook.com for any custom tool modification requests.

Enables a check against Update set name change, customer updates changes, reopening of update sets. Note: It is recommended to keep this property turned ON and to educate the developers about not manipulating update sets once they are closed. The tool functioning might be affected if developers perform any of the above mentioned action. ?

☐ Yes | No

Enter the instance which will orchestrate all the commits. Ideally, this would be the production instance but it can also be a non-prod instance. This property will be useful in a non-production environment to ensure that the scheduled job which orchestrates all the commit do not accidentally run in an environment that is not designated as the orchestrating environment. (Ensure that all non-production environments have the same value for this specific property where this tool is installed) ?

This property enables application logging for debugging purposes in the non orchestrator instance. Logs can be found in the syslog table in the respective non prod instance with source as 'service_guru_auto_deployment'. ?

☐ Yes | No

Save

- a) Turn on the below property (highly recommended) to enable the best practice check for update sets in your non-production instance.

Enables a check against Update set name change, customer updates changes, reopening of update sets. Note: It is recommended to keep this property turned ON and to educate the developers about not manipulating update sets once they are closed. The tool functioning might be affected if developers perform any of the above mentioned action. (?)

☐ Yes | No

- b) Define the Orchestrator instance name.

Enter the instance which will orchestrate all the commits. Ideally, this would be the production instance but it can also be a non-prod instance. This property will be useful in a non-production environment to ensure that the scheduled job which orchestrates all the commit do not accidentally run in an environment that is not designated as the orchestrating environment. (Ensure that all non-production environments have the same value for this specific property where this tool is installed) (?)

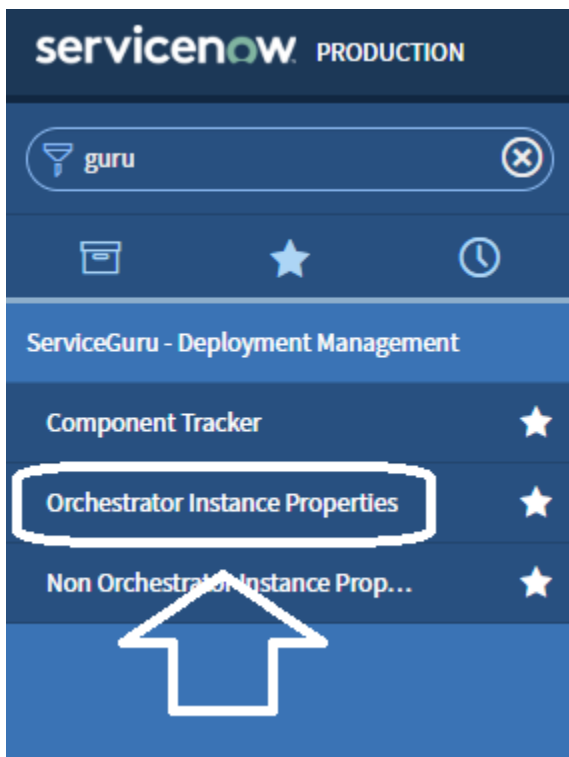
- c) Enable logging in the Non-orchestrating instance related to script execution and update sets

This property enables application logging for debugging purposes in the non orchestrator instance. Logs can be found in the syslog table in the respective non prod instance with source as 'service_guru_auto_deployment'. (?)

☐ Yes | No

Orchestrator instance installation steps

- 1) Install the batch update sets (update set name: ServiceGuru-Shell) in all the instances which are in scope for the integration.
- 2) Type 'ServiceGuru' in the left navigation and click on 'Orchestrator Instance Properties' properties.



- b. You should see the configuration page like the one below

Important: The changes to properties on this page are local to this instance. You will need to perform the same property changes in respective ServiceNow instances where this tool is installed.

Contact: shettyravish@outlook.com for any custom tool modification requests.

Should we Turn ON the integration? (This property is an ON/OFF Switch for this Integration.) (?)

☐ Yes | No

This property enables application logging for debugging purposes. Logs can be found in the syslog_app_scope table. (?)

☐ Yes | No

This property enables sending of digest emails every 2 hours (emails are only sent if Application logging is enabled). (?)

☐ Yes | No

Select the environments which need change request for any deployments Note: Time zone to evaluate the change window is as per the timezone of the user profile which runs the Scheduled job. (?)

None ▼

Custom Change Request Query: (WARNING: Perform tests to verify if the query is correct or else it would cause the tool to fail in execution.) Enter the Encoded Query for Change Request table which can be used to query the Components. (?)

active=true ⓘ

Choose the mode of dealing with when Warnings and Errors occur. Any changes to this property would be across all environments and you cannot pick and choose different behavior for different Releases. In Manual mode, the errors and warnings will have to be resolved manually by the developers and the tool will wait till it no longer sees any Warnings or Errors. In 'Smart Accent' mode similar

- c. In the below properties, update the Non-orchestrator instances Username and password field with the integration account created in previous steps.

Shared Non Orchestrator Instance credentials (Username) - This property sets the username for the User Profile used for Integration. This same Username and Password combination is used across all Non-Orchestrator environments. (?)

Shared Non Orchestrator Instance credentials (Password) - This property sets the password for the User Profile used for Integration. This same Username and Password combination is used across all Non-Orchestrator environments. (?)

..... ⓘ

- d. In the below properties, update the Production Username and password field with the integration account created in previous steps.

Production Only - This property sets the username for the User Profile used for Integration for the Production instance defined in the property `x_7756_update_set.service_guru_production_instance`. [?](#)

Production Only - This property sets the Password for the User Profile used for Integration for the Production instance defined in the property `x_7756_update_set.service_guru_production_instance`. [?](#)

e. Enter the Operations/Reviewers group name

ServiceNow Group name of Operations team who will get emails and Tasks in workflows for Update Source fixes (Only 1 team name can be accommodated) [?](#)

f. Select the environments which will require a change ticket for the deployment

Select the environments which need change request for any deployments Note: Time zone to evaluate the change window is as per the timezone of the user profile which runs the Scheduled job. [?](#)

None

None

Production only

All environments except QA/Dev

All environments except Dev

g. Update the production environment name here. To test this tool in the lower environment, you can select a non-production environment instead of production.

Enter the production instance name. [?](#)

h. Enter the name of the group which will get the digest emails

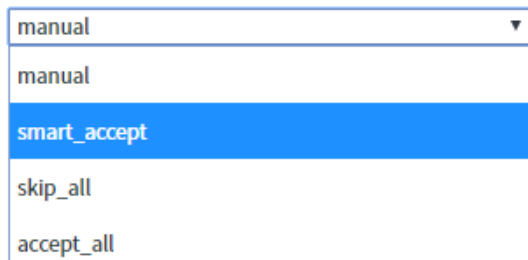
ServiceNow Group name who will get digest emails (Only 1 team name can be accommodated) [?](#)

i. Update the names of all environments that will be part of the integration. Ensure that there is no space and comma is used as the delimiter.

Enter all non orchestrator environments which will be part of this integration. Use a comma as the delimiter to separate each entry (example: dev1,dev2,uat1,uat2). (?)

- j. Change the 'Error/Warning' acceptance mode as desired. By default, it is selected as 'manual'. You can change it to 'accept all', 'skip all' or 'smart accept'.

Choose the mode of dealing with when Warnings and Errors occur. Any changes to this property would be across all environments and you cannot pick and choose different behavior for different Releases. In Manual mode, the errors and warnings will have to be resolved manually by the developers and the tool will wait till it no longer sees any Warnings or Errors. In 'Smart Accept' mode, similar to manual mode, the errors and warnings need to be resolved by the developers based on their first occurrence. For subsequent instances, the tool will be accept/skip based on the decisions made in the previous environment if the same kind of collision is found. 'Skip all' would ignore all errors and warnings. 'Accept all' would accept all errors and warnings irrespective of the type of errors and warnings. (?)




Deployment? (Toggle approvals for the Request workflow which is for the always evaluated for production environment defined in the property irrespective of value of this property. (?)

- k. Enter the release table name which you use (for example rm_release). You can also use any task table. Also, define the release query to identify which release is valid.

Release Table Name (If you do not have the release management plugin activated, create a custom task table and update this property with the name of the property) (?)

Custom Release Query: (WARNING: Perform tests to verify if the query is correct or else it would cause the tool to fail in execution.) Enter the Encoded Query for Release table which can be used to query the Components. (?)

- l. Once all the configurations are completed, turn on the below property. You should be all set.

Should we Turn ON the integration? (This property is an ON/OFF Switch for this Integration.) 

☒ Yes | No

- 4) Open the catalog item 'ServiceGuru - Code Promotion' in the maintain item (sc_cat_item) table. Control the visibility of this catalog as required. Also, tag this catalog to appropriate categories.