

# State Flows

---

# Introduction

---

## State Flows

---



**Note:** This article applies to Fuji. For more current information, see *State Flows* <sup>[1]</sup> at <http://docs.servicenow.com> The ServiceNow Wiki is no longer being updated. Please refer to <http://docs.servicenow.com> for the latest product documentation.

### Overview

State flows enable an administrator to customize transitions from one state to another in tables derived from the Task [task] table and configure the system to perform work during transitions to specific states. An example of a state transition is when the **State** field in an incident changes from **Active** to **Awaiting User Info**. An administrator might want to trigger an event during this transition or make a specific field mandatory when the incident reaches the end state.

State transitions in the Work Management application were reimplemented to use state flows. For information about customizing Work Management state flows, see *State Flow Customization*.

The state flows feature is available starting with the Eureka release.

### How State Flows Work

When you create a new state flow, the system automatically replaces the usual programming elements that control task states with customizable business rules, client scripts, and UI actions that give you a wide range of processing options. You can configure custom state transitions to occur automatically or manually and in any order. You can create business rules and UI actions that only appear and take effect for certain states.

- **Manual transitions:** Initiate manual transitions by a UI action that the system creates when you provide a condition or a script.
- **Automatic transitions:** Initiate automatic transitions from a business rule that the system creates when you provide a condition and a script.

### Available with State Flows

- **Custom transitions:** Customize the order in which states can change for records in any table that extends the Task [task] table.
  - **Field controls:** Control the behavior and visibility of specific fields when a task changes states or reaches a specified end state.
  - **State choice list:** Limit the values offered in a task record's **State** field to valid states for that transition. This is the same client script that the system creates to manage field controls for state transitions.
  - **Events:** Trigger events when a state transition occurs or when a record reaches a specific end state.
-

## Start and End States

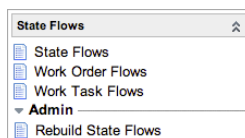
You can create a custom state flow for processing that must occur when a task record makes a specific transition from one state to another. These records require a starting state and an ending state, and processing occurs during the transition between states. To perform some processing when a task record reaches a particular end state, you only need to define the end state. In some cases a state flow can have a starting state only, such as when you need to perform some type of cleanup after a task is canceled. A state flow might have no starting or ending state if the processing in the record applies to more than one state transition. The solution is to store the business rule or client script in a state flow record and create a condition to trigger processing for any state change that requires it. An example of this in work management is the Roll Up Changes business rule on the Work Order Task [wm\_task] table. This business rule rolls up state changes that occur in tasks to the parent work order.

## Roles

Users with the admin role can create, edit, and delete state flow records.

## Menus and Modules

Activating state flows adds the State Flow application to the application navigator, with the following modules (starting with the Eureka release).



- **State Flows:** Shows all state flows for task-based tables. Work order and work order task state flow records are *not* displayed unless Work Management is activated.
- **Work Order Flows:** Shows all state flows defined for work orders. This module is available when Work Management is activated.
- **Work Task Flows:** Shows all state flows defined for work order tasks. This module is available when Work Management is activated.
- **Admin**
  - **Rebuild State Flows:** Updates links to the states in a state flow record to repair damaged state flows.

## Activating State Flows

An administrator can activate the State Flows plugin. This plugin is activated automatically when Work Management is activated.

**Click the plus to expand instructions for activating a plugin.**

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
  - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click **Activate**.

## References

[1] [https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/state-flows/concept/c\\_StateFlows.html](https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/state-flows/concept/c_StateFlows.html)

# Installed with State Flows

---

## Overview

The following components are installed with State Flows:

- Tables
- Script Includes
- Business Rules

State flows are available starting with the Eureka release.

## Tables

State flows add the following table.

Table	Description
State Flow [sf_state_flow]	Contains state flow definitions. This table contains all state flow definitions, including those for work orders and work order tasks.
Work Order Flow [sf_work_order]	Contains state flow definitions for work orders. This table is installed when Work Management is activated.
Work Task Flow [sf_work_task]	Contains state flow definitions for work order tasks. This table is installed when Work Management is activated.

## Script Includes

State flows add the following script includes.

Name	Description
StateFlow	Implements state flows and supports creation of state flow elements, such as business rules, UI actions, dictionary overrides, and client scripts.
StateFlowAJAX	Allows access to state flow functionality from client scripts.

## Business Rules

State flows add the following business rules.

---

Name	Table	Description
Assert Field Uniqueness in State Flow	State Flow [sf_state_flow]	Ensures that business rules and UI actions are not accidentally copied to new state flows.
Check Client Script	State Flow [sf_state_flow]	Adds a client script to new records.
Check Event Rule	State Flow [sf_state_flow]	Adds or deletes event rules, as the event field is updated.
Check Work Notes Rule	State Flow [sf_state_flow]	Adds or deletes work note rules, as the work notes for a state flow are updated.
Create Business Rule	State Flow [sf_state_flow]	Automatically creates a business rule when automatic conditions or script are present.
Create script for Field controls	State Flow [sf_state_flow]	Create scripts for field controls, when they are in use
Create UI Action	State Flow [sf_state_flow]	Automatically creates a UI action when manual conditions or script are filled in.
Delete Related Elements	State Flow [sf_state_flow]	When state flows are deleted, delete all related client scripts, business rules, UI actions and overrides.
Remove script for Field controls	State Flow [sf_state_flow]	If all field controls are disabled, see if any of the client scripts should be removed.
State Change	State Flow [sf_state_flow]	Get the correct state choice value when the state is changed.
Update dependent records	State Flow [sf_state_flow]	When a state flow is made active or inactive, ensure the business rule and UI actions are made active or inactive as well.

---

# Using

---

## Using State Flows

---

### Overview

You can create custom state flows for any table in ServiceNow that uses states. Make state transitions occur in any order or skip specific states according to your business practices. State flows can manage the behavior of fields and field visibility on a task form. State flows can also trigger system events that perform work such as sending email notifications.

Administrators can configure state flows to:

- Limit the choice list for the **State** field to contain only those states that are valid for the flow.
- Define the behavior and visibility of specific fields on a task form when state transitions occur and for certain end states.
- Configure events that are triggered when a task changes states.
- Set up UI actions and business rules to implement certain state transitions, or while the task is in certain states.

The State Flow plugin installs records used by work management to control state transitions for work orders and work order tasks. For detailed information about using work management state flows, see [State Flow Customization](#).

Creating and customizing state flows requires scripting knowledge. Users with the admin role can create state flow records.

### Creating State Flows

1. Navigate to **State Flows > State Flows** and click **New**.
2. Fill in the fields, as appropriate (see table).
3. Determine if you want the state transition method to be manual or automatic and open the appropriate section on the form.
  - **Manual:** Click **Create UI Action** to create a button on the task form that enables users to execute the transition manually. The system uses the value in the **Name** field as the label for the UI action. The UI action executes the script in the **Manual Script** field when the conditions are true. For example, a manual transition can create an **Activate** button when an incident is in the **New** state that enables a user to mark the incident as *active*.
  - **Automatic:** Click **Create Business Rule** to create the business rule. The business rule executes the script in the **Automatic Script** field when the conditions are true. For example, a business rule created by the system can set an incident state to **Assigned** when the **Assigned to** field is populated. Business rules are automatically deleted when the state flow record is deleted. For more information, see [Business Rules](#).
4. [Optional] Click **Create Client Script** to create the script that limits the values offered in a task record's **State** field to valid states for that transition.
5. Configure the fields in the **Field Controls** section to control how specific fields display when a task record changes states.

The system enforces the field controls with the same client script that filters the choice list for the **State** field.

---

Field	Description
Number	Record number automatically generated by ServiceNow.
Table	[Required] Table on which the state flow record runs. Only tables that extend the Task [task] table are available in the list.
Starting state	Name of the state at the beginning of the transition. The selections in this field are filtered by the possible states for the table selected. For more information, see Start and End States.
Ending state	Name of the state at the end of the transition. The selections in this field are filtered by the possible states for the table selected.
Client script	Client script to run for this transition. The client script controls the available states you can select by limiting the contents of the <b>State</b> choice list to valid states. This client script also controls specific field behavior configured for state changes in the Field Controls section of the form.
Event	Name of an existing event to trigger when this transition occurs. See Triggering Events on State Changes for more information.
Name	[Required] Name of this record. Make sure the name is descriptive of the state transition or the processing that the record is performing. This name does not have to be unique.
Roles	Not used for any processing.
Active	Enables or disables this state flow record.
Class	[Required] Defines the state flow class for this record. The system selects the appropriate class from these options: <ul style="list-style-type: none"> <li>• <b>State Flow:</b> Records created for state flows in all task-based tables except those in work management.</li> <li>• <b>Work Order Flow:</b> Records created for state flows in the Work Order [wm_order] table. This class is available when work management is activated.</li> <li>• <b>Work Task Flow:</b> Records created for state flows in the Work Order Task [wm_task] table. This class is available when work management is activated.</li> </ul>
Dictionary override	Sets the starting value for the <b>State</b> field on all new records for the table named in the state flow record. See Dictionary Overrides for configuration procedures.
Work notes	Noteworthy comments about this state flow transition. For details about how these notes are used, see Work Notes.
Comment	Details about the customized record.

**Manual** (Runs scripts from a UI action that require the user to click a button or related link.)

Manual condition string	Conditions for enabling a UI action that cannot be defined with the condition builder. For example, you can use this string to define UI actions for mobile devices. This condition has an <b>[and]</b> relationship with the condition in the <b>Manual condition</b> field.
Manual condition	Conditions for enabling a UI action that can be defined for fields in the target table. This condition has an <b>[and]</b> relationship with the condition in the <b>Manual condition string</b> field.
Manual script	Script that defines what the UI action does when the conditions are true. This script runs when the user clicks a button or a related link.
UI action	[Read Only] Name of the button that the system creates to enable this transition. The system creates the label using the same name as the state flow record that created it.

**Automatic** (Runs a business rule automatically when a task record is changed and updated.)

Automatic condition string	Conditions for running the business rule that cannot be defined with the condition builder, such as evaluating if the proposed transition is a valid flow. This condition has an <b>[and]</b> relationship with the condition in the <b>Automatic condition</b> field.
Automatic condition	Conditions for running the business rule that can be defined for fields in the target table. This condition has an <b>[and]</b> relationship with the condition in the <b>Automatic condition string</b> field.
Automatic script	Script that performs additional work when the condition is true. This script can do tasks such as update the date and time the transition occurred or notify someone using email when a specific state change occurs. Automatic state transitions occur when changes are made to the task record.
Business rule	Name of the business rule created for this transition. Two conditions must be satisfied before this business rule can run. The task must be on a specific starting state, and the <b>Automatic condition</b> must be true. If both of these conditions are satisfied, the business rule performs the transition requested, using the starting and ending states from the State Flow form.

**Field Controls** (Determines field properties when a record transitions between states or reaches a specific end state.)

Mandatory fields	Makes the selected fields required when this transition occurs, or when the end state is the current state.
Read only fields	Prevents the selected fields from being edited when this transition occurs, or when the end state is the current state.
Visible fields	Displays the selected fields when this transition occurs, or when the end state is the current state.
Not mandatory	Makes the selected fields optional when this transition occurs, or when the end state is the current state.
Not read only	Makes the selected fields editable when this transition occurs, or when the end state is the current state.
Not visible	Hides the selected fields when this transition occurs, or when the end state is the current state.

## Dictionary Overrides

A dictionary override in a state flow defines the starting state for all new records in a specific table. You set an override in tables that extend a base table only, so that your customizations are applied only to the extended table.

1. In a state flow record, select an **Ending state**.

This is the override value which becomes the starting state for all new records in the table named.

2. Click **Create Default Value**.

The system populates the **Dictionary override** field with a value of **state**, which is the field in the task table affected by the override. The **Dictionary override** field is read-only. After the override is created, the system hides the **Create Default Value** button on all subsequent state flow forms for that table.

## Work Notes

Work notes are an important part of the state flow process and are used to communicate information about state transitions. The state flow adds these work notes to the **Work notes** field of any task making this transition.

These rules apply to state flow work notes:

- For a state flow with no **Starting state**, the work note is added every time the task transitions to the **Ending state**.
- For a state flow with a **Starting state** and an **Ending state**, the work note is added only when the task transitions from that starting state to that ending state.
- If two state flows with work notes have the same **Ending state**, but only one has a **Starting state**, the system adds the work notes from the state flow with the starting state. This better matches the state flow work note to the more important transition between specific starting and ending states.

## Field Controls

You can define controls for individual fields that are enforced when a record transitions between states. Settings in the Field Controls section of the State Flow form enable you to apply field controls when the system detects a specified state transition or when the end state is the current state when the form is opened. The control is applied only to *existing* fields on the form. State flows cannot add fields to the form.

For example, you might want the **Problem** field to be visible when an incident moves to the **Awaiting Problem** state. If the incident state changes to **Awaiting User Info**, you hide the **Problem** field and make the **Caller** field mandatory.

The best practice when creating field controls is to configure state flow records with an ending state only and to create the correct behavior for every ending state you want to control. This ensures that the field controls are set properly when the user selects a new state, and also when the user returns a record's **State** field to the original state. Only specify a full state transition, with both a starting and ending state, when you want a particular behavior for that precise state transition.





**Note:** State flows use client scripts to enforce field controls. It is possible that your settings can be changed by existing UI policies, which execute after client scripts.

ServiceNow creates the following objects as needed to enforce field properties in state flows:

Type	Name	Description
Business rule	State Flow Notes for <table name>	Enforces mandatory fields for the table on which that field behavior is defined.
Client script (onLoad)	<table name> state flow	Sets possible states and initial mandatory, read-only, and visible properties when a record is loaded.
Client script (onChange)	<table name> change state flow	Sets updated mandatory, read-only, and visible properties when a record is changed.

## Events Triggered on State Changes

You can configure a state flow to trigger a registered system event when a task transitions from a starting state to a specified end state. For example, you can use events to trigger email notifications and create script actions. When you attach an event to a state flow, ServiceNow creates a business rule called **State Flow Events for <table name>** for the table specified in the state flow. If you specify a start and end state, the business rule executes when the record transitions from the start state to the end state. If the state flow only specifies an end state, the business rule executes whenever that end state is reached. The system creates one business rule for all state flows containing events on a single table. When all events or all state flows on a table are deleted, the system deletes the business rule.

## Rebuild State Flows

When you use an XML file to import a state flow record into an instance, the system attempts to match the incoming states with existing states by comparing sys\_ids. Because the sys\_ids of items in a choice list can vary between instances, the system can fail to match the states, even though they are otherwise identical. When matching fails, the start and end states of affected records are left blank or contain numeric values. To repair these records navigate to **State Flows > Admin > Rebuild State Flows**. This module runs a script that compares the numerical value of each item in the **State** field choice list until it finds a match in the imported state flow record.

## State Flow Cleanup

The business rules, client scripts, and UI actions that the system creates automatically to perform custom transitions exist only while the state flow records that use them are present. When all the state flows on a table are deleted, the system attempts to delete any unnecessary programming elements that were created on that table, using these criteria:

---

Element	Deleted When
UI action	The state flow that created it is deleted.
Business rule	
Dictionary override	
Business rule that processes events triggered by a state flow	All state flows for the table specified that have events configured are deleted.
Client script (onLoad)	All state flows for the table are deleted.
Client script (onChange)	All state flows with field controls are deleted.
Work notes business rule	All state flows with field controls or work notes are deleted

---

# Article Sources and Contributors

**State Flows** *Source:* <http://wiki.servicenow.com/index.php?oldid=250372> *Contributors:* Cheryl.dolan, Emily.partridge, John.ramos, Steven.wood

**Installed with State Flows** *Source:* <http://wiki.servicenow.com/index.php?oldid=228251> *Contributors:* Emily.partridge, Steven.wood

**Using State Flows** *Source:* <http://wiki.servicenow.com/index.php?oldid=234377> *Contributors:* Cheryl.dolan, Emily.partridge, Steven.wood

# Image Sources, Licenses and Contributors

**Image:Warning.gif** *Source:* <http://wiki.servicenow.com/index.php?title=File:Warning.gif> *License:* unknown *Contributors:* CapaJC

**Image:State\_Flow\_Menu.png** *Source:* [http://wiki.servicenow.com/index.php?title=File:State\\_Flow\\_Menu.png](http://wiki.servicenow.com/index.php?title=File:State_Flow_Menu.png) *License:* unknown *Contributors:* Steven.wood