

# ServiceNow Glossary

The ServiceNow Developer Glossary defines the ServiceNow terms of interest to developers

---

## ServiceNow Developer Glossary

The ServiceNow Developer Glossary defines terms that apply to the Now Platform.

### A

***Access [App Engine Studio]***: A tab in the Catalog editor used to select which users or groups can access a Catalog Item.

***Access Control List (ACL)***: A group of *Access Control* rules applied to a resource.

***Access Control rule***: Controls the data that users can access and the access operations, such as create or update, that are allowed. *Access Control* rules require users to pass a set of requirements to gain access to particular data.

***action [Flow Designer]***: A reusable operation used in Flow Designer flows or subflows.

***action (mobile)***: Encapsulates action items into buttons on mobile screens.

***Action Designer***: The Flow Designer environment for creating and editing actions. Use Action Designer to add actions to a flow.

***action item***: Defines a database operation allowed on a table's records in mobile application screens. Action items can be used to create, update, or delete a record.

***action item parameter***: Enables users to enter values for fields when invoking action items in mobile applications. Most action items use user-supplied values rather than hard-coded values to set field values at runtime.

***Action name***: Defines a name to use when referencing a UI Action in scripts.

***action parameter mapping***: Maps action item parameters to UI parameters. UI parameters prompt users to enter values. Action parameter mappings map user-entered values to parameters in action items.

***action.sendRedirectURL()***: A method used to specify the page to open after a UI Action executes.

***action.setReturnURL()***: A method used to specify the page to open when users click the back button on a redirect page after a UI Action executes.

***action step***: An action step, or step, is a single reusable operation within an action. Action steps require subject matter expertise with application tables, fields, and business logic. Application developers or IT generalists add action steps to actions in the Action Designer design environment.

***activate [Flow Designer]***: Enable a flow to cause the flow to execute in the runtime environment when trigger conditions are met.

**activity [Workflow]:** Workflow blocks that perform different tasks, such as obtaining approvals, sending email, running scripts, testing conditions, and setting field values on records.

**admin:** A user with the *admin* (administrator) role in ServiceNow has permission to perform almost all functions. The term "admin" usually includes both the global admin and the admin for a given application. For example, both the admin and catalog\_admin roles have rights to administer the service catalog.

**Agent Workspace:** a suite of tools that provides agents, case managers, help desk professionals, and managers quick access to what they need to help answer customer questions and resolve customer problems.

**Analytics Hub:** The user interface for assessing, comparing, and predicting the performance of Performance Analytics indicators over time. Analytics Hub displays data for a single indicator.

**API Explorer:** See *REST API Explorer*.

**applet [Mobile]:** Users access mobile application data through mini applications called applets.

**applet launcher:** A page of related applets that a user accesses from the mobile navigation bar.

**applet parameters:** Provide values to data item parameters at runtime. Applet parameter values are typically entered by users but can be hard-coded.

***application:*** A packaged set of configuration records that provides a business solution. Developers use the Now Platform to build and manage custom applications.

***Application Explorer:*** The navigation pane on the left side of the Studio screen that displays a list of application files.

***application file:*** Configuration records that enable developers to extend application functionality. Application developers create application files when they add configuration records for application logic such as business rules, workflows, and script includes.

***Application menu:*** A grouping of modules as they appear in the Application Navigator (UI16) or *All* menu (Next Experience). Users may refer to an Application menu as simply an application.

***Application Navigator:*** The navigation pane on the left side of the ServiceNow screen in UI16 that displays a searchable list of applications and modules, user favorites, and recently visited pages.

***application picker:*** Enables application developers to view and select the application where their changes apply. The application picker is available from the Developer pane of the System menu.

***application properties:*** Configurable application parameters to change an application's behavior. Typically, only application administrators can read and write application properties. Application properties are records in the sys\_properties table.

***application repository:*** See *ServiceNow application repository*.

***application scope:*** A unique namespace and a set of permissions to access application files and data. Developers can specify the data that is accessible to other applications from the application record or from each table record.

***assert type:*** The Assert type field specifies the conditions that must be met for an Automated Test Framework test to pass. Test designers can use assertions to specify whether the results of an operation are expected or unexpected. For example, if you wanted to verify that a record cannot be updated, you would add a Record Insert test step and set the *Assert* type field to **Record was not inserted**. The test would pass when the record insert failed.

***automation task:*** See *scheduled job*.

***Automated Test Framework (ATF):*** Enables you to create and run automated tests to confirm that your instance works after making a change. Examine failed test results to identify the changes that caused the failure and the changes that you should review.

## B

***base class:*** A table that is extended but does not extend any other tables.

***binding:*** See *data binding*.

***branch:*** A source control term for work in progress on an application that is separate from the released application code.

***breakdown:*** Groups or filters indicator scores for more detailed analysis and groups the scores by the breakdown elements.

**breakdown element:** The values for a breakdown.

**breakpoint [JavaScript Debugger]:** Pauses script execution in the JavaScript Debugger to allow examining variables and their values during runtime.

**bucket group:** Gathers data into groups when the breakdown element for grouping scores is not available in a table in the application. For example, a bucket group might be an hourly chart showing how busy a store is throughout the day.

**business rule:** Server-side logic or JavaScript configured to run when a record is inserted, updated, or deleted, has occurred. In the case of a query, the business rule should run before the database operation so that the data returned to users is appropriate to their system privileges (user roles). A typical business rule might execute after a user updates an incident or escalates the priority of a change request.

- **before:** Execute business rule logic before performing the database operation.
- **after:** Execute business rule logic immediately after the database operation.
- **async:** Execute business rule logic on a different processing thread in near real time after the database operation. When the execution occurs depends on the scheduled job scheduler.
- **display:** Pass the display object, g\_ scratchpad to the calling client-side script.

## C

***canvas [Virtual Agent]***: The work area for creating a topic flow in Virtual Agent Designer.

***canvas [Workflow]***: The work area for creating workflows in the Workflow Editor.

***catalog [Service Catalog]***: A section of the service catalog where users can order items and services.

***catalog client scripts***: Scripts for adding dynamic effects and validation to service catalog forms. These scripts apply to service catalog items or variable sets. (Standard client scripts use tables.)

***Catalog Item [Service Catalog]***: A form that describes a good or service a user can order in the service catalog.

***catalog UI policy [Service Catalog]***: Controls to specify which fields are visible when viewing catalog items, catalog tasks, and requested item tasks.

***category [Service Catalog]***: A bucket to organize catalog items.

***chart***: A visual representation of data in a report or on a form.

***class (script)***: Some script types, such as script Includes and installation exits, create or extend JavaScript classes and can be instantiated in other script types.

***class (table)***: A table that extends or is extended by another table.

**client script:** Client-side JavaScript that runs in a browser.

ServiceNow supports the following script types:

- **onLoad()** runs when a form is loaded.
- **onChange()** runs when a form's field value changes.
- **onSubmit()** runs when a form is saved, submitted, or updated.
- **onCellEdit()** runs when a cell value on a list changes.

**client script [UI Builder]:** Client-side JavaScript that interact with components and client state parameters on a page. Client scripts are mapped to events as event handlers in UI Builder.

**client state parameter [UI Builder]:** The use of client state parameters, events, and client scripts to add interactivity to a page.

**client state parameter [UI Builder]:** Page variables that can be used to configure components on a page and modified via events to make a page dynamic.

**client test runner:** A browser tab that runs client-side test steps in Automated Test Framework. Client test runners require a browser tab to function. If no client test runner is available when you run a test, you are prompted to open one. Testers can manually start a client test runner or select an existing client test runner. Test designers can schedule starting client test runners when they schedule running a test suite.



***coalesce***: The method by which import sets may update existing records in a destination production table rather than simply inserting new records. When coalescence is configured for a field, the import set application attempts to match the source value to the current target value on a production table for updating. If the source value does not match, a record is created.

***collision***: A collision occurs when data being imported into a ServiceNow table conflicts with a record already in the target table.

***commit***: The act of making changes part of the working branch of a source control repository. Add a commit message to describe the committed changes.

***component library [Now Experience UI Framework]***: Reusable building blocks for creating custom components.

***component playground [Now Experience UI Framework]***: An interactive tool, on the ServiceNow Developer Site, for configuring library components

***conditional breakpoint [JavaScript Debugger]***: Pauses script execution in the JavaScript Debugger when a condition script evaluates to true to allow examining variables and their values during runtime.

***configuration item (CI)***: A physical entity, such as a computer or router; a logical entity, such as an instance of a database; or a conceptual entity, such as a requisition service. CIs can be managed and tracked through Configuration Management and stored in the Configuration Management database (CMDB).

***conflict [source control]***: A conflict occurs when a single application file is updated by multiple developers concurrently.

***connection & credential alias***: A pointer to a connection definition and credential record to use in IntegrationHub steps.

***connection attribute***: A connection-specific variable to pass values to IntegrationHub steps.

***Console Debugger [JavaScript Debugger]***: Console window that allows expression evaluation in real time when paused at a breakpoint. Evaluation happens in the scope, context, and thread in which execution is paused.

***container [Service Portal]***: Divides a Service Portal page into sections. A container can contain between one and six columns of widgets.

***content item [Service Catalog]***: A record that provides information in the service catalog.

***context***: See *workflow context*.

***context menu***: Controls for a list or form specific to the selected table or record.

***control [Virtual Agent]***: An operation used in Virtual Agent topic flows.

***conversation [Virtual Agent]***: An instance of a topic flow created when a user initiates a session with Virtual Agent.

**core action:** An action in ServiceNow that is available to any flow that cannot be viewed or edited from the Action Designer design environment.

**credential:** A record that stores the details needed to authenticate to an external platform.

**cross-origin resource sharing rules (CORS):** Control which domains can access specific REST API endpoints in ServiceNow.

**cross-scope privilege:** Specifies whether access is allowed to cross-scope resources. Cross-scope resources are application files which exist in different application scopes.

**current update set:** An update set that has been selected using the Update Set Picker. User customizations will be added to the user's current update set.

**custom application:** Any scoped application a developer creates on the Now Platform.

**customizable record:** A record that is tracked by update sets.

**customization:** A change to a customizable record.

## D

**dashboard:** A collection of visualizations grouped into a single location.

**data binding [UI Builder]:** The process of associating data with a UI element that displays the information.

***data broker [UI Builder]:*** See *data resource*.

***data collector:*** A scheduled job that collects data from indicator sources.

***data item:*** Filtered sets of data from a table. Data items determine the information that is available to an applet.

***data item parameters:*** Data item parameters create variables that work with applet UI parameters to allow users to dynamically filter the data sent to an applet.

***data lookup:*** A platform feature that sets one or more field values when certain conditions are met.

***data lookup definition:*** A control to automatically change field values based on data lookup rules.

***data model:*** The table structure an application uses and the relationships between the application tables.

***data pill:*** A visual representation of a variable in Automated Test Framework, Flow Designer, and Virtual Agent.

***data pill picker:*** An interface to select a data pill related to a test, flow, action, subflow, topic flow, or related table.

***data policy:*** Enforces data consistency by setting mandatory and read-only states for fields. Data policies can apply rules to all data entered into ServiceNow, including data brought in through import sets, web services, data entered through the mobile UI, or any process that inserts or updates data. Data policies can also act like UI policies.

***data resource [UI Builder]***: A dynamic, reusable way of defining what data to fetch for a page's components.

***data resource instance [UI Builder]***: A configured implementation of a data resource within an application.

***data stream action***: IntegrationHub action for interacting with REST or SOAP web services that return responses larger than 10 MB or return paginated responses.

***data source***: A data source can be from a file or a Java Database Connectivity (JDBC) connection. Used to create an import set so that data can be processed, if necessary, prior to being mapped and imported into a target table. Data sources support the following remote file retrieval methods: FTP, FTPS (Auth SSL, Auth TLS, Implicit SSL, Implicit TLS), HTTP, HTTPS, SCP. They can be a file or a Java Database Connectivity (JDBC) connection.

***declarative action event mapping [UI Builder]***: A scriptless way to use events to add custom functionality to forms, fields lists, and other components.

***delegated development***: Allows designated users without a system admin role to develop or deploy applications on the Now Platform.

***Destination [App Engine Studio]***: A tab in the Catalog editor used to specify the table that will store the data captured in a Catalog Item.

***Details [App Engine Studio]***: A tab in the Catalog editor used to configure the *Basic info* and *Item details* for a Catalog Item.

***developer pane***: Group of settings in the *System* menu (gear icon in the ServiceNow banner) through which developers configure their system setting preferences.

***dialog***: See *modal*.

***dictionary***: A table, sys\_dictionary, that describes the tables and fields in an instance. It stores field data types, character limits, default values, dependencies, and other attributes.

***dictionary override***: A declaration in a dictionary record for an extended table that overrides the dictionary entry inherited from the parent table.

***domain***: A unique section of a single instance used to separate data, processes, and UI elements while still sharing global properties and global processes across the entire instance.

***domain hierarchy***: The relationship structure between domains in a domain-separated instance. The hierarchy is created by the parent-child relationships between domains. Members of a domain see only data contained within their domain or the child domains that are lower in the domain hierarchy.

***domain scope***: Sets the domains a user can access in a domain-separated environment.

***domain separation***: Enables developers to separate data, processes, and administrative tasks into logical groupings called domains. Developers can then control the information available to each domain, including which users can see and access the data. Typically used only in Multiple Service Provider (MSP) organizations.

***dot-walking***: Describes the structure of a variable that contains one or more reference fields, separated by a dot for each table involved.

## E

***ECC (External Communication Channel) Queue***: A database table that is normally queried, updated, and inserted into by external systems such as the MID Server used for Discovery. The ECC Queue is the normal connection point between ServiceNow and external systems. Records saved to the Queue [ecc\_queue] table are in the form of messages from external systems.

***email template***: Reusable subject line and message body content for email notifications.

***Embedded Help***: A Now Platform feature that enables developers to add context-sensitive help to their applications. Users can choose whether to view the help.

***Embedded Help qualifier***: Determines the query parameter values to use when determining which Embedded Help page to display.

***embedded list***: A related list on a form that acts like a field instead of as a related list.

***entity view action mapper [UI Builder]***: Also known as EVAM. Standardizes the format for displaying data in cards and lists.

***EVAM***: See *entity view action mapper*

***EVAM Datasource [UI Builder]***: Specifies which data sources (tables) and which records to send to user experience components.

**EVAM View Config [UI Builder]:** Defines which records and table fields are sent by an EVAM Data Resource to a component.

**EVAM View Config Action Assignment [UI Builder]:** Specifies what to do when a user clicks a card in a user experience component.

**EVAM View Config Bundle [UI Builder]:** A collection of EVAM View Configs.

**EVAM View Template [UI Builder]:** Defines the layout of a card and the position of fields and values on the card.

**event:** A configured marker in JavaScript that triggers a notification or script action. Events are generated in business rules or other server-side scripts. Developers can use existing system events or create their own.

**event [UI Builder]:** An action a user takes or occurrence that happens on a page.

**event handler [UI Builder]:** An action performed when an event occurs.

**event mapping [UI Builder]:** The process of identifying an event handler to run when an event occurs.

**event log:** A module that collects information when events are generated. This information includes when the event was generated, the name of the event, the values passed in the event parameters, and when the event was processed.

**event registration:** The act of adding an event to the event registry.



**event registry:** The table of events that ServiceNow processes respond to.

**execution plan:** Simple, linear automated processes. Execution plans are deprecated. For new processes, use flows (Flow Designer) or workflows.

**Experience [App Engine Studio]:** A graphical interface users interact with to create workspaces, portals, catalog items, and applications accessible through a ServiceNow native mobile app.

**export definition:** Determines the data to include in an export set.

**export set:** Determines the data to export and the export target to use when exporting data.

**export target:** Specifies the target file on a MID server to which the export set data will be written.

**extension instance:** A registered instance of a scripted extension point that links a script include with a scripted extension point. The script include can be called whenever the Scripted Extension Point is invoked.

- **client extension instance:** A registered instance of a client extension point that links a UI script with a client extension point. The UI script is included on pages that invoke the client extension point.
- **UI extension instance:** A registered instance of a UI extension point that links a UI macro with a UI extension point. The macro can be called whenever the UI extension point is invoked.

**extension point:** Designates where custom script logic can be incorporated into your code so you can integrate customizations and new features without altering the existing code for your application. The returned data or objects must conform to the requirements specified by the application creator.

- **client extension point:** Used in Jelly pages (UI pages) to include UI scripts in a page without needing to specify all of the script URLs in advance. Typically, you will define the functionality that implementations if the extension point should provide and include the appropriate Jelly tag in your UI page. Upon rendering the page, all registered implementations of your client extension point will be included.
- **UI extension point:** Runs one or more UI macros when rendering a UI page or calling a UI macro containing a UI extension point. Provides "spots" in custom pages and macros where additional UI features and functionality can be included without having to update the original page or macro.

## F

**field:** A field contains an individual item of data on a record in a table.

**field status indicator:** A colored symbol or vertical bar that might appear to the left of form elements. These indicators convey field status, such as mandatory, read-only, or modified. The appearance of field status indicators varies depending on the version of the UI in use.

**field watcher:** Tool for tracking and displaying all actions that the system performs on a selected form field. Administrators can use the field watcher to determine what happens to the field and how the

value of the field changes when an event takes place. Administrators can also impersonate non-admin users to debug what happens when those users make changes on an instance.

***filter navigator:*** Search field in the Application Navigator (UI16) and *All* menu (Next Experience) used to filter the list of modules and applications as you type. Can also be used to open forms and lists using shortcuts like *incident.do* or *incident.LIST*.

***fix script:*** Server-side JavaScript that runs after a custom application is installed or upgraded. Fix scripts can also be run on demand. Developers include fix scripts in applications to make changes that are necessary for data integrity or application stability.

***flow:*** An automated process consisting of a sequence of actions and a trigger. Flows automate business logic for a particular application or process.

***Flow Designer:*** A Now Platform feature for automating processes in a single design environment. Flow Designer enables process owners to automate approvals, tasks, notifications, and record operations.

***flow execution details:*** Flow execution details provide runtime information about an action or flow directly from the design environment, such as the current state, actions or steps run, and values produced. Open related records from embedded Now Platform editors or in a new tab.

***form:*** A content page that displays fields and values for a single record from a database table.

***form layout:*** The arrangement of fields on a form.

***function [Mobile]:*** Functions that define UI elements that determine what a user can do in a ServiceNow mobile application. The three types of functions are:

- **Actions:** Change data, such as modifying a field value.
- **Navigation:** Move between screens, such as opening a record from a list.
- **Smart buttons:** Enable users to take an action on demand, such as making a telephone call.

## G

***Git integration:*** An integration that enables developers to save and manage multiple versions of an application in a Git source control repository from non-production instances.

***Glide:*** An extensible Web 2.0 development platform written in Java that facilitates rapid development of forms-based workflow applications (work orders, trouble ticketing, and project management, for example).

***glide list:*** An array of values, often referenced from another table, that are stored in a single field (for example, the Watch List field in the task table).

***GlideForm (g\_form):*** GlideForm is the JavaScript class used to customize forms. This class contains the g\_form object. GlideForm and the g\_form object only run on the client.

***GlideRecord class***: A JavaScript class used to interact with the ServiceNow database from a script.

***GlideScriptable class***: A JavaScript class that replaces a Java Packages call.

***global UI script***: A script available on any form in the system. Administrators can create global UI scripts.

***group***: A set of users who share a common purpose.

***Guided Application Creator (GAC)***: A wizard-based user interface for creating custom applications on the Now Platform. To modify existing applications, use Studio.

***Guided Tour***: A pre-scripted demonstration of how to use a custom application. Guided tours contain a series of interactive steps to help users complete tasks with a custom application.

## H

***handled event [UI Builder]***: A page-level, developer-created event that is mapped to component events.

***hello world***: A simple application commonly used as a first development exercise.

## I

***import set***: A set of data imported into ServiceNow from an external data source. Imported data is stored in the temporary Import Set table, where a transform map maps inserts or updates the data in existing tables.

***import set web service***: A web service that transforms incoming data synchronously based on the associated transform maps.

***inbound email action***: A control that specifies how to process email received by a ServiceNow instance.

***indicator source***: Defines sets of records from a ServiceNow table that have a common characteristic, for example, that the Priority field value is critical. Indicators use indicator sources to calculate scores. Indicator scores are KPI values. Indicator sources specify a table and the frequency for collecting data.

***instance***: An individual implementation of ServiceNow. Each instance has a unique address (usually <http://<InstanceName>.service-now.com/>). Many customers have multiple instances, such as sandbox, development, and production instances.

***integration***: A process by which the ServiceNow platform can be made to work with a third-party application or web service.

***IntegrationHub***: A framework used to interact with third-party platforms in ServiceNow.

***interactive***: A transaction generated by a user as opposed to by a background or system process.

***item parameter [Mobile]***: See *data item parameter*.

## J

***Java***: The ServiceNow application is written in Java. Java was first release in 1995 and is still widely used today in enterprise-level applications.

**JavaScript:** A programming language used by ServiceNow for code that is executed within the browser as well as server-side code. See *Rhino*.

**JavaScript Debugger:** Application for debugging business rules and other synchronous server-side scripts.

**JavaScript Log:** A tab in the JavaScript Log and Field Watcher application for viewing `jslog()` and `console.log` messages written by client-side scripts.

**JS include:** Client-side JavaScript that can be associated with a widget and used in the widget client controller.

## K

**keyword [Virtual Agent]:** Words or phrases that a user enters to initiate a conversation about a topic. Keywords are not used when Natural Language Understanding (NLU) is enabled for Virtual Agent.

## L

**list:** A content page that displays zero or more records from a database table. The list has rows and columns. Each row is a record, and each column is a field from the record.

**list control:** Settings to specify which features are available on a list, such as the New and Edit buttons.

**list layout:** The arrangement of columns in a list.

***Location [App Engine Studio]***: A tab in the Catalog editor used to assign the catalog and category where a Catalog Item will be located in Service Portal.

***logpoint [JavaScript Debugger]***: Write debugging information to the JavaScript Debugger's Session Log without editing the server-side script being debugged.

## M

***many to many definition***: A many-to-many relationship allows multiple records from one table to be related to multiple records in another table. A many-to-many definition describes this relationship and configures an intermediate table to create the relationship.

***merge [source control]***: The process to move changes from one branch of a source control repository to another.

***message***: A control to specify the string to display in the user interface for a specific language.

***metric***: A measure to evaluate the effectiveness of a process.

***metric definition***: A record used to configure the calculation used for a metric.

***MID (Management, Instrumentation, and Discovery) Server***: The MID Server is a service that runs on a local network to enable communication between a ServiceNow instance and external applications, data sources, and services on the local network.



***mobile experience [Mobile, App Engine Studio]:*** A version of a ServiceNow application created in App Engine Studio that is accessed from the ServiceNow Agent mobile application on supported iOS and Android devices

***modal:*** User experience window that overlays another content window and takes control of the user experience.

***module:*** Any link in the Application Navigator (UI16) or *All* menu (Next Experience) that opens a page in the content frame or in a separate tab or window.

## N

***natural language understanding (NLU):*** A platform capability that allows ServiceNow to understand user statements.

***navigation bar [Mobile]:*** A widget at the bottom of every mobile app screen that enables users to navigate to applets, settings, and notifications. For custom mobile applications, developers must add an applet launcher tab to the navigation bar.

***node [Virtual Agent]:*** An instance of a Virtual Agent control on the Virtual Agent Designer canvas configured for a topic flow.

***notification:*** Message sent from a ServiceNow instance to inform a user that something happened.

***Now Experience CLI (CLI):*** Command line tool for developing custom components using the Now Experience UI Framework.

**Now Experience UI Framework:** Enables developers to build custom web components to create a modern user interface for application users.

## O

**offline mode [Mobile]:** Allows ServiceNow Agent mobile application users to access application records from a local cache when no Internet connection is available.

**order guide:** A Service Catalog item that steps a user through the process of ordering multiple related catalog items as a single request.

## P

**personal developer instance (PDI):** A free, fully featured ServiceNow instance available to Developer Program members that provides a sandbox environment for them to develop applications or improve their skills on the Now Platform.

**platform:** The complete set of the ServiceNow software services.

**plugin:** An extension to an existing platform that is activated to add functionality to the platform. Many plugins can be activated by the administrator but some require assistance from ServiceNow Customer Support. Plugins cannot be uninstalled.

**portal:** See *Service Portal*.

**portal page:** See *Service Portal page*.

***preview script usage [REST Message]***: A related link in a REST Message record that generates a server-side JavaScript code stub to invoke the API.

***process plan***: The information necessary to execute a flow such as the sequence of published actions or subflows, the input values for each subflow or action, the action steps to run for each action, and the data provided by the trigger or subflow output.

***processor***: HTTP endpoints available to perform arbitrary scripting actions.

***project [Now Experience UI Framework]***: Structure that organizes component files.

***publish (application)***: The act of posting an application to the organization's application repository so that it can be installed on other instances in the organization.

***publish [Flow Designer]***: The act of making an action or subflow available to flows.

***publish [Virtual Agent]***: The act of making a topic available for Virtual Agent conversations.

***publish [Workflow]***: The act of making a workflow active for all users on an instance.

***pull request [source control]***: A pull request indicates that the committed changes on a development branch are ready for review. A pull request begins the process to merge a branch.

## Q

**qualifier:** See *Embedded Help qualifier*.

**Questions [App Engine Studio]:** A tab in the Catalog editor used to add questions to a Catalog Item. Questions in a Catalog Item capture information from users that is needed to fulfill a request. Questions are grouped into types and subtypes that configure how users should answer the questions.

**quick start test:** A ServiceNow-provided test or test suite installed with the demo data of an application that can be used as templates to build your own tests and test suites.

## R

**range [Reporting]:** Defines data intervals that are used in bar and pie charts.

**record producer:** A catalog item that allows end users to create records from the Service Catalog.

**record watch [Service Portal]:** A listener for a widget to receive notifications for changes made to records in user interfaces other than Service Portal. Uses the `spUtil.recordWatch()` method.

**reference key:** Used to identify a field other than `sys_id` to use as the unique identifier for a reference field. The value of the reference key field, instead of the `sys_id`, is stored in the database for that reference field.

**related links:** UI actions at the bottom of a list or form to run a script on one or more records.

***related list***: A group of records from a single table at the bottom of a form that have a relationship with the record on the form.

***relationship***: A connection between two tables in a database based on a reference field or a scripted query.

***report***: A representation of data to organize, summarize, and present data to convey information in a meaningful way.

***repository [ServiceNow application]***: See *ServiceNow application repository*.

***repository [source control]***: A collection of application files and metadata that includes changes over time and commit details.

***repository configuration***: The configuration information used to connect an application to a source control repository.

***restricted caller access (RCA)***: The Restricted Caller Access (RCA) feature enables an administrator to define cross-scope access to an application or application resource and allow or deny access requests.

***REST API Explorer***: A ServiceNow tool that uses information from your instance to provide a list of endpoints, methods, and variables that you can use to build and send REST requests. After you build the request, the REST API Explorer provides code samples in multiple programming languages that you can use to initiate the request, and detailed request and response information.

***REST message***: A record that stores details on how to interact with an external web service through REST.

**Rhino:** An open-source implementation of JavaScript written entirely in Java. It is typically embedded into Java applications to provide scripting to end users.

**role:** A category that can be assigned to a group or user, and that can be used to grant access to particular parts of ServiceNow. Once access has been granted to a role, all groups or users assigned to that role are granted the same access. Roles can also contain other roles; access granted to a role will be granted to any roles that contain it.

## S

***scheduled data export:*** Specifies a schedule when export sets will be run. A single export can be scheduled or regular intervals can be scheduled with support for including only delta records.

***scheduled data import:*** Specifies when to import data from data sources. Transform maps are applied to the imported data before writing the data to the target table.

***scheduled email of report:*** Defines the frequency at which reports are run and the results sent as PDF files by email to recipients.

***scheduled job:*** A record that performs a given task based on a schedule. Scheduled jobs can generate and distribute reports, generate records from templates, or run a script.

***scheduled script execution:*** A record that executes a server-side script based on a schedule.

***scope:*** See *application scope*.

**screen UI policy:** Client-side logic that controls which fields are visible or mandatory on mobile app screens.

**script action:** Server-side JavaScript that is executed when a particular event is generated.

**Script Debugger:** Tool for debugging server-side JavaScript through the use of breakpoints.

**script include:** Reusable, server-side JavaScript that executes on request. Each script include defines either an object class or a function.

**script tree:** A collapsible pane on the side of a script field that can be used to insert fields, methods, and properties into a script.

**scripted REST API:** Enables defining service endpoints, query parameters, and headers for a scripted web service API, as well as scripts to manage the request and response. Listed in the Scripted REST APIs [sys\_ws\_definition] table.

**scripted web service:** Enables developers to create their own APIs on the Now Platform. Third-party applications use scripted web services to access records in ServiceNow tables.

**Scripts - Background:** Module for executing server-side JavaScript on demand. Developers can specify the scope to use when executing the code.

**server script:** A JavaScript script that is processed by the server.

**service:** An application or feature that performs activities in support of either business applications or the Now Platform.

**Service Creator:** Enables a department to offer custom services through the service catalog, for example, an HR department offering tuition reimbursement for further education. Each published service has an associated record producer catalog item. Users designated as managers and editors create and design these catalog items. End users can request services by ordering the catalog item.

**Service Portal:** A ServiceNow user interface (UI), built using the Service Portal framework, that provides an alternative user experience to the standard UI. It enables users to interact with the underlying Now Platform using a minimum number of clicks from any device: desktop, tablet, or smartphone.

**Service Portal page:** Houses containers, rows, and columns, which then contain widgets. By designing the layout of the page and the widgets within it, developers construct the desired user experience. Pages are reusable across portals.

**ServiceNow application repository:** Location for customers to publish their custom applications. Published applications can be installed on any of the customer's corporate instances.

**ServiceNow Studio:** Provides an Integrated Development Environment (IDE)-like interface for application developers to create, review, and update application files in one centralized location from a tabbed environment.

**session debug:** Displays debugging messages in the user interface. When enabled, session debugging is active during the user session or until disabled. Activate using the System Diagnostics application.



***session log [JavaScript Debugger]***: Tab in the JavaScript Debugger for viewing transactions and other session-related information. Developers can select which data to view.

***Settings [App Engine Studio]***: A tab in the Catalog editor used to configure the behavior of a Catalog Item. Creators can adjust the *Portal settings* to customize the checkout area of a Catalog Item.

- **Hide 'Add to wishlist'**: Select to hide the *Add to wish list* button on the Catalog Item.
- **Hide attachment**: Select to hide the *Add attachments* button on the Catalog Item.
- **Make attachment mandatory**: Select to make adding an attachment mandatory for a Catalog Item.

***Share***: A section of developer.servicenow.com where Developer Program members can upload or download applications, scripts, or other application artifacts for free.

***slushbucket***: A slushbucket is a two-column interface for choosing multiple selections from an available pool of items.

***SOAP Message***: A record that stores details on how to interact with an external web service through SOAP.

***source control***: The management of code and changes to the code of an application.

***spoke***: A scoped application containing Flow Designer content dedicated to a particular application or record type.

***stash***: Changes to an application saved locally without committing to a source control repository. Developers can stash work, work on a different branch, and then reapply the changes captured in the stash

later.

**state [Now Experience UI Framework]:** The properties and property values for a custom component.

**step:** See *action step*.

**step configuration [ATF]:** A specific test action that the Automated Test Framework can run. Step configurations do not contain any runtime test data and can be run only when test designers add them as part of a test step. Each step configuration is a record in the Test Step Config [sys\_atf\_step\_config] table, which specifies the input variables used to run the step configuration and the output variables available to other step configurations.

**step result:** Stores the output of a test step run. Each step result is a record in the Step Result [sys\_atf\_test\_result\_step] table that specifies the status of the test step, a summary of the output, and a complete log of the output generated by the test step. Use step results to identify failures and functionality needing review.

**step variable:** Stores step-specific input and output values. Use step variables to specify a particular test step target or to pass information to other test steps.

**Studio:** See *ServiceNow Studio*.

**style:** A control to specify the appearance of a particular field.

**style sheet [Service Portal]:** A CSS record that can be added to any Service Portal page or component to overwrite the theme.

**subflow:** An automated process consisting of a sequence of reusable actions and specific data inputs that allow the process to be started from a flow, subflow, or script. Subflows automate generic business logic that can be applied to multiple applications or processes.

**sys\_id:** Each record in ServiceNow is identified by a unique 32-character GUID (Globally Unique ID) called a sys\_id. The same sys\_id value is never generated twice, ensuring every record created in every table in every instance of ServiceNow in the world has a unique identifying value.

**system property:** See *application property*.

**system property category:** Groups application properties into sections on a page.

## T

**table:** Related data held in a table format within a database. A table is made up of columns and rows.

**table column:** A set of data values, all of a single type, in a table. Columns define the data in a table.

**table row:** A single structured data item in a table. Rows, also known as records, are used to populate data into a table.

**table transform map:** See *transform map*.

**tag [source control]:** A fixed set of committed application files from a source control repository.

**target [export]:** The file on a MID Server where exported data will be written.

**target [import]:** The table defined in a transform map to receive imported records.

**template:** Predefined values to populate a form with record data.

**test [ATF]:** A logical grouping of related automated test steps that verify some functionality or feature. Each test is a record in the Test [sys\_atf\_test] table. Test designers typically create a test to verify one feature or a group of related features.

**test result:** Stores the output of a test or test suite run. Each test result is a record in the Test Result [sys\_atf\_test\_result] table that specifies the duration of the test run, the status of the test, and screenshots where available. Use test results to identify failing or non-running tests. Use the test logs to see more information about test results. By default, the system deletes test and test suite results 30 days after creation unless you enable the option to retain the test result indefinitely.

**test step:** Combines a step configuration with the runtime test data necessary to run a step. The test step always specifies the order in which it runs in the test. Test steps have their own related list of step results. Each test step is a record in the Test Step [sys\_atf\_step] table that specifies a test action, the step configuration, and an execution order. Test designers add test steps to tests to verify functionality.

**test suite:** A collection of tests that run in a specific order. Test designers typically create a test suite to test an application or a group of related features. Test designers can schedule running test suites

and starting any required client test runners.

***theme [Service Portal]***: CSS defined in the CSS variables field in the Themes table [sp\_theme] that sets the appearance of a portal and portal widgets.

***topic [Virtual Agent]***: A blueprint that defines the dialog between Virtual Agent and a user to accomplish a goal. A topic has topic properties and a topic flow.

***topic flow [Virtual Agent]***: The component of a topic that describes the possible paths that a user's conversation with Virtual Agent can take.

***topic properties [Virtual Agent]***: The configuration values for a property that identify the name and description for the topic along with who can interact with a topic and when the topic should run.

***tracked customizations***: Customizations to tables that are recorded in update sets.

***transaction***: Any foreground process requested by the user (URL request).

***transaction quota***: Specifies when to cancel a transaction. The system cancels any transaction that matches the conditions of a transaction quota rule.

***transform map***: A set of field maps that determine the relationships between fields in an import set and fields in an existing ServiceNow table, such as Incident [incident] or User [sys\_user]. After creating a

transform map, you can reuse it to map data from another import set to the same table.

***trigger [Flow Designer]***: The record, date, or application conditions that are met to initiate the execution of a flow.

## U

***UI Action***: An interactive component on a form or list that performs a specific action. UI actions include the buttons, links, and context menu items on forms and lists.

***UI Action Visibility***: A related list on UI Action forms used to restrict the view of UI Actions using include and exclude rules. If there are no rules configured for a UI Action, the UI Action will appear in all form views.

- ***Include***: Rule that allows UI Actions to display on specified form views.
- ***Exclude***: Rule that prevents UI Actions from displaying on specified form views.

***UI Builder (UIB)***: A WYSIWYG web user interface builder. UI Builder enables developers to build new pages or customize existing pages for Agent Workspace and portals using Now Experience UI Framework components.

***UI formatter***: A form element used to display information that is not a field in the record. You add formatters to a form by configuring the form.

**UI macro:** A discrete scripted component that administrators can add to the user interface. UI macros are typically controls that provide inputs or information not provided by existing field types.

**UI page:** A collection of custom HTML and script content that displays UI components. Creating UI pages requires a knowledge of HTML or Jelly. You can also create simple AngularJS applications using UI pages.

**UI policy:** Dynamically changes the behavior of information on a form and controls custom process flows for tasks. For example, you can use UI policies to make a field on a form read-only, make a field mandatory, and hide other fields. Basic UI policies do not require any scripting; however, for more advanced actions, use the Run scripts option.

**UI script:** A reusable client-side JavaScript script that changes the behavior of UI components. Administrators can run UI scripts from client scripts, from other client-side script objects, and from HTML code in UI pages.

**update set:** A collection of customizations made to a customer instance that can be transferred to another instance. When an instance is customized, the change is added automatically to the current update set. There can be only one current update set per user at a time.

**update set batch:** A group of update sets related by the *Parent* field. Used to move multiple related update sets between instances.

**update set picker:** A choice list in the instance header bar that enables the user to select the update set that stores customizations the user makes.

**update source:** A ServiceNow instance that the current instance can retrieve update sets from.

## V

**validation script:** A client script that validates field values as users enter them. When a field value fails validation, the script produces an error and prevents users from saving the record.

**variable set:** A group of variables that can be shared between catalog items.

**variant [UI Builder]:** A version of a UI Builder page, such as a 404 page, with access controlled by role or condition. Under differing conditions, users see different versions of the page.

**view [database]:** Defines table joins for reporting purposes.

**view [form or list]:** An alternate layout for presentation of a record's data. Different user profiles use different views to see data from the same record.

**view [Now Experience UI Framework]:** JSX function describing what a component looks like when rendered.

**view rule:** When a user switches form or list views, the selected view is saved as a user preference so the user sees the same view by default when the form or list opens. View rules override this functionality to force a specified view to be used.



**virtual agent:** A conversational bot platform that provides user assistance through conversations in a messaging interface.

## W

**web service:** Allows diverse applications to communicate with each other. ServiceNow supports both inbound (provider) and outbound (consumer) web services.

**widget [Performance Analytics]:** A Performance Analytics widget determines how data is presented on dashboards. Configure widgets to view, set up, edit, and manage properties for dashboards and visualization types: time series, scores, lists, and breakdowns.

**widget [Service Portal]:** A Service Portal widget is an interactive component in the Service Portal user interface. You can define what a widget does using HTML templates, CSS, client scripts, server scripts, and JavaScript dependencies. From an AngularJS standpoint, widgets are a superset of an Angular directive.

**widget dependency:** Links JavaScript and CSS files to widgets to create dependencies between widgets and third-party libraries, external style sheets, and angular modules.

**widget editor:** The application for editing Service Portal widget components.

**widget option schema:** Defines the user-configurable fields for a Service Portal widget, for example, how many records a list widget displays.

***widget options***: Developer-settable parameters that enable each Service Portal widget instance to be uniquely configured.

***workflow***: A sequence of activities to automate processes in applications.

***workflow context***: A workflow's runtime environment. The runtime environment, once launched, is unaffected by workflow version changes.

***workflow editor***: The interface used to visually define a workflow.

***workflow schedule***: A record that configures a workflow to run on a schedule.

***workflow version***: A record that stores the workflow configuration when published or checked out. Only one published workflow version can be active at a time. Only one checked out workflow version can exist at a time.