



ServiceNow

# **ServiceNow**

## **Fundamentals**

participant guide





# ServiceNow Fundamentals

---

## Table of Contents

<b>Module 1: User Interface and Navigation .....</b>	<b>7</b>
Lab 1.1: ServiceNow Overview.....	22
Lab 1.2: Lists and Filters .....	46
Lab 1.3: Forms and Templates .....	66
Lab 1.4: Branding .....	83
 <b>Module 2: Users and Tasks.....</b>	 <b>89</b>
Lab 2.1: User Administration .....	98
Lab 2.2: Task Management .....	118
Lab 2.3: Notifications .....	134
 <b>Module 3: Data Administration.....</b>	 <b>141</b>
Lab 3.1: Data Schema.....	155
Lab 3.2: CMDB.....	171
Lab 3.3: Import Sets .....	183
Lab 3.4: Reporting .....	204
Lab 3.5: Data Security .....	221
 <b>Module 4: Service Automation .....</b>	 <b>229</b>
Lab 4.1: Knowledge Management .....	237
Lab 4.2: Service Catalog .....	253
Lab 4.3: Workflows .....	268
Lab 4.4: Service Level Agreements (SLAs) .....	286
 <b>Module 5: Introduction to Scripting and Development .....</b>	 <b>291</b>
Lab 5.1: Scripting.....	307
Lab 5.2: Performance and Upgrades.....	327
Lab 5.3: Development .....	341



service**now**

1

2

3

4

5

## Module 1

### User Interface and Navigation

- 1.1 ServiceNow Overview
  - 1.2 Table Lists
  - 1.3 Forms
  - 1.4 Branding
-

## Module 1

service*now*

### 1.1 ServiceNow Overview

- What is ServiceNow?
- Cloud Infrastructure
- Multi-Instance Software Stack
- Key Platform UI Components
  - Content Frame
  - Banner Frame
  - Application Navigator
- Mobile Access
- Product Documentation

### **"ServiceNow is changing the way people work"**

A leader in Enterprise Service Management (ESM), ServiceNow is a cloud-based platform offering Platform as a Service (PaaS) that contains a number of modular applications that can vary by instance and user, automating common business processes

"When I started ServiceNow, my vision was to build a cloud-based platform that would enable regular people to create meaningful applications to route work through an enterprise."

- Fred Luddy, ServiceNow Founder

The ServiceNow Service Automation Platform provides a modern, easy-to-use, service management solution in the cloud allowing your organization to automate manual repetitive setup tasks, manage your core IT processes, standardize service delivery, and focus on your core business, not just ITSM infrastructure.

ServiceNow provides all of this to users from a configurable web-based user interface, built on top of a flexible table schema.

The ServiceNow platform and the applications that run on it use a single system of record and a common data model to consolidate your organization's business processes.

Another advantage to this single system is that you can leverage it to build your own custom applications.

The ServiceNow platform provides a PaaS, a cloud-based computing model that provides the infrastructure needed to develop, run, and manage applications.

## ServiceNow Cloud Infrastructure

service*now*



ServiceNow provides a true Enterprise cloud, meaning it is not limited to a specific department or function but encompasses the entire enterprise

There is even more to the ServiceNow platform, and that is the enterprise cloud; the computing environment that delivers software, infrastructure, and platform services to your business.

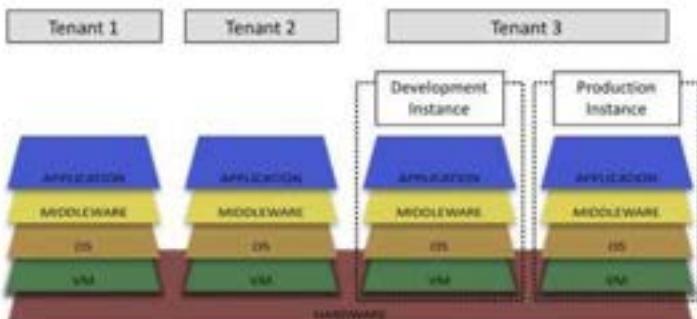
## Multi-Instance Software Stack

service<sup>now</sup>

The ServiceNow cloud is built on an advanced multi-instance, single tenant architecture

In this architecture, all data, applications, and customizations reside in a unique software stack called an instance

Each tenant (or customer organization) may have more than one instance



Many Software as a Service (SaaS) or Platform as a Service (PaaS) systems utilize a multi-tenancy cloud architecture as their default for all customers. This means that customer data is co-mingled on a single database.

They rely on large and complex databases that require hardware and software maintenance on a regular basis, resulting in availability issues for customers, and any action that affects the multi-tenant database affects all shared customers.

ServiceNow utilizes an advanced, multi-instance, single tenant architecture as the default offering for customers.

An instance features an individually isolated database containing data, applications, and customizations.

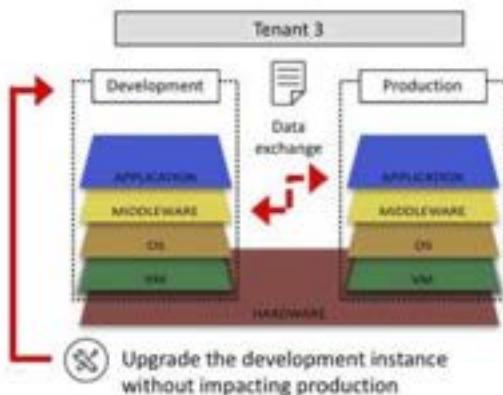
## Multi-Instance Software Stack: Benefits

servicenow

Within the organization's software stack, instances are isolated but can still communicate with each other

Data can be exchanged between instances in a number of ways

Upgrades can be made on individual instances



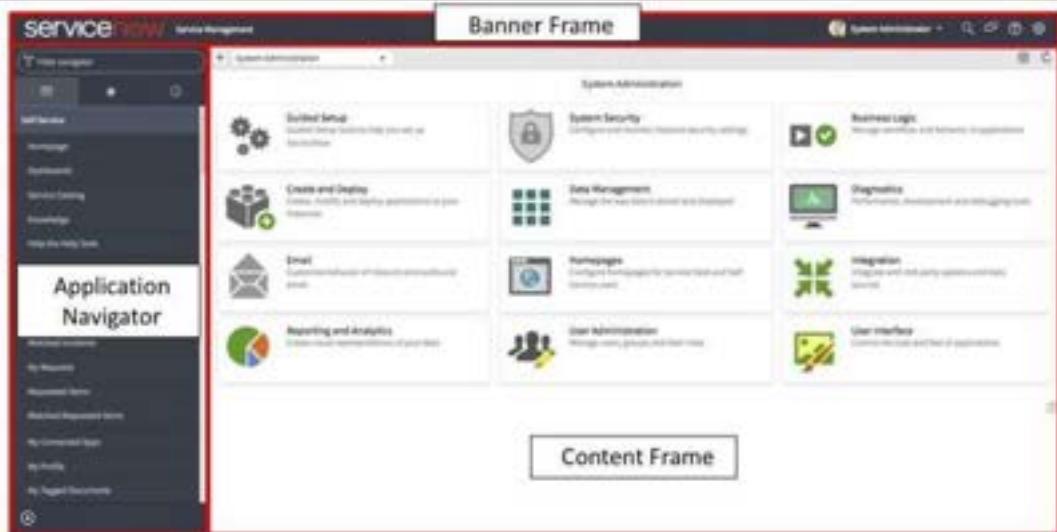
Each instance is isolated from every other instance, but they can still communicate with each other.

The ServiceNow multi-instance architecture provides these distinct advantages:

- The multi-instance architecture allows ServiceNow to perform actions on individual customer instances such as performing an upgrade, on a schedule that fits the compliance requirements and needs of your enterprise.
- Data is truly isolated in their own databases, making hardware and software maintenance on these unique customer instances far easier to perform and issues can be resolved on a customer-by-customer basis.

## Key Platform UI Components

service<sup>now</sup>



The User Interface (UI) is the main way to interact with the applications and information in a ServiceNow instance. Notable ServiceNow features include real-time form updates, user presence, an application navigator designed with tabs for favorites and history, and enhanced activity streams all of which you will explore in this training. This is an example of the System Administration homepage.

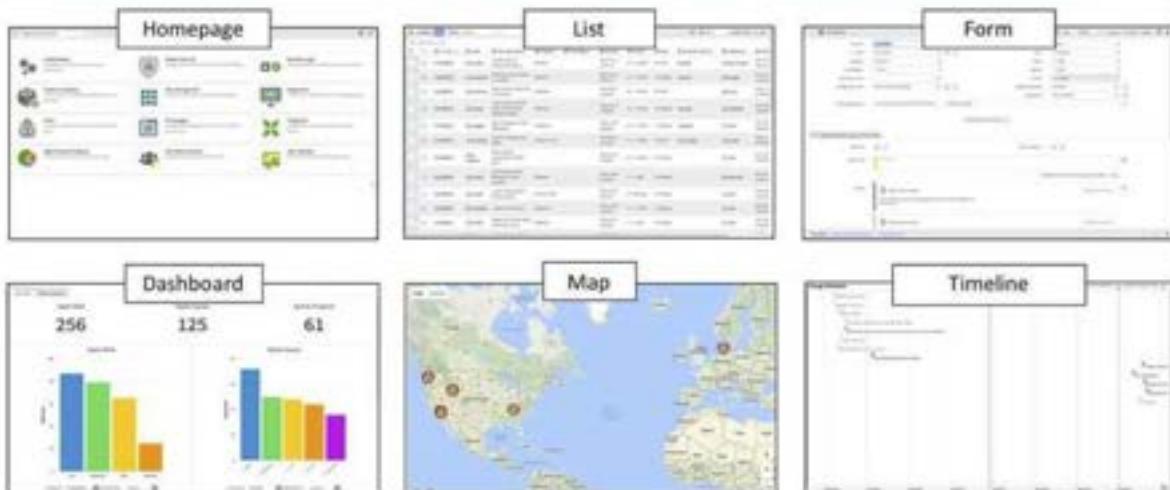
The ServiceNow user interface is divided into three areas:

- 1. Banner Frame:** The Banner Frame highlights important tools and settings that apply to your instance.
- 2. Application Navigator:** The components of the Application Navigator, the panel on the left side, are based upon your assigned role(s). The navigator may be expanded (as shown above) or collapsed. The navigator provides links to all application menus and modules, based on your permissions.
- 3. Content Frame:** The Content Frame displays information, such as lists, forms, dashboards, knowledge bases, and service catalogs depending on where you navigate within the platform. This also impacts how the information is visually represented.

**NOTE:** The position of these components on your screen may vary depending on your region.

## Content Frame: Common Types of Interfaces

service<sup>now</sup>



**Homepage:** A homepage consists of navigational elements, functional controls, and platform information. When a user logs in to an instance, the default homepage defined for their role appears unless the user switched to another homepage or has set a dashboard to appear.

All users with a role can use the **Add content** link on the homepage to customize the homepage and display important changes and emergency information to other users.

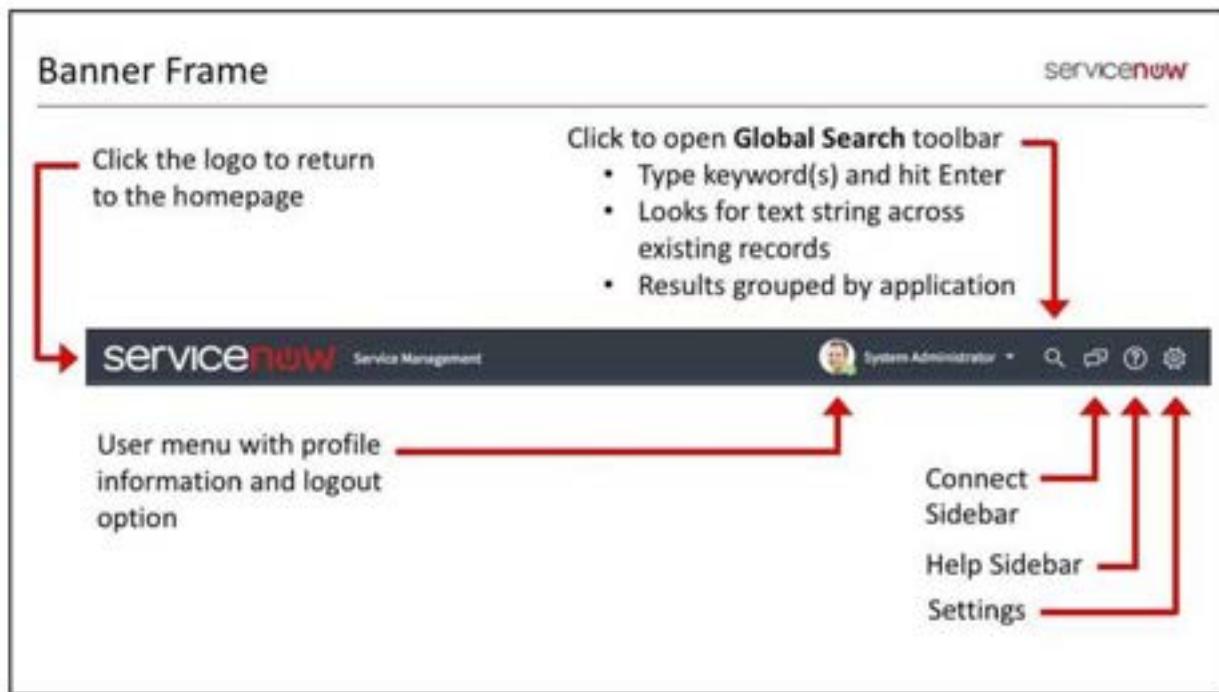
**List:** View data records as a list. Lists display records from a data table, as well as allow you to edit the record information using the List Editor functionality.

**Form:** View individual data records as a form. Data is typically entered into ServiceNow through forms.

**Dashboard:** Dashboards enable you to display multiple performance analytics, reporting, and other widgets on a single screen.

**Map:** Display ServiceNow data graphically on a Google map. Drill-down into a map to view specific data points.

**Timeline:** Used to track tasks or projects.



The Banner Frame runs across the top of every page and contains global navigation controls and several key functionalities and features:

- Your logo in the top-left hand corner, which also navigates you back to your homepage when you click on it
- Information about the logged-in user: click the down arrow to the right of the user name to view the user profile or log out
- Click the magnifying glass to expand the Global Search toolbar and use this to search across all data in ServiceNow, such as a keyword, record number, and more
- Toggle on and off the Connect Sidebar, which is used to communicate with other users in real-time
- Get help, including Product Documentation and new features
- Personalize your settings

**NOTE:** With additional rights, a user may see Impersonate User and Elevate Roles as additional options from the user menu.

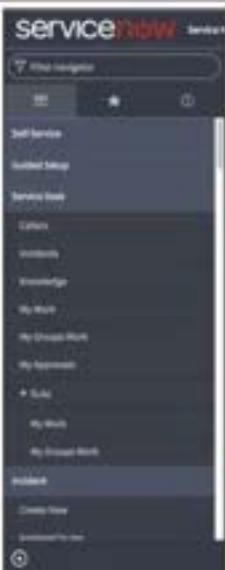
## Application Navigator

servicenow

The Application Navigator is a list of available applications and their corresponding modules

Application Menu

Applications are a collection of modules for a specific process like Incident or Change



Modules enable navigation to different areas of the platform, including:

- Links to a new record
- Filtered lists of records
- Special view pages

Applications are a group of modules, or pages, that provide related information and functionality in an instance. Modules can contain links to a new record, lists of records with varying filters applied, and special visual tools.

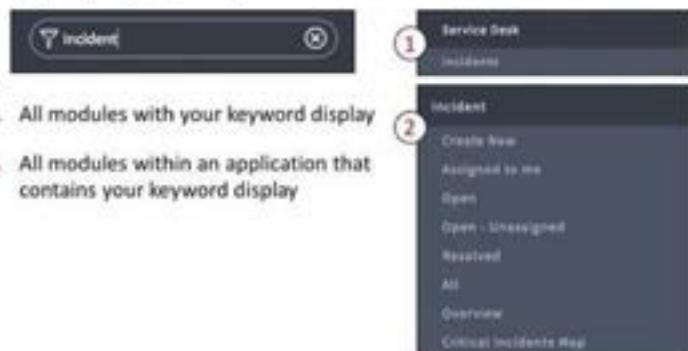
For example, the Incident application contains modules for creating and viewing incidents. The Configuration application contains modules for changing and accessing servers, databases, and networks.

## Application Navigator: Filtering

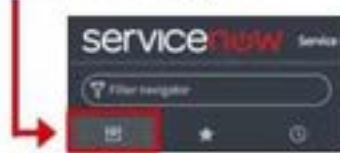
servicenow

Above the application list, use the **Filter Navigator** to quickly navigate to applications and modules

Simply begin typing the Application or Module name



Use the **All Applications** icon to view all applications within the navigator



The Application Navigator provides access to all applications and the modules they contain, enabling users to quickly find information and services.

To view all applications within the navigator, ensure that the **All Applications** icon is selected at the top left of the navigator.

**TIP:** Double-click the **All Applications** icon to expand and/or collapse all applications. Click any application to expand or collapse all of its modules.

To quickly search throughout your application navigator to view a particular application or module, use the **Filter Navigator**. The Filter Navigator is located at the top of the Application Navigator.

As soon as you begin typing, the Application Navigator displays only applications and/or modules matching your keyword. For example, if you type "Incident" into the Filter Navigator, you will view the Incident application and a list of *all* its modules, as well as any modules containing the word "Incident" within other applications, such as **Service Desk > Incidents**.

## Application Navigator: Favorites and History

servicenow

### Favorites

Items you add as favorites appear in the **Favorites** tab of the Application Navigator, which is represented by a star icon

Favorites appear as icons in a collapsed view

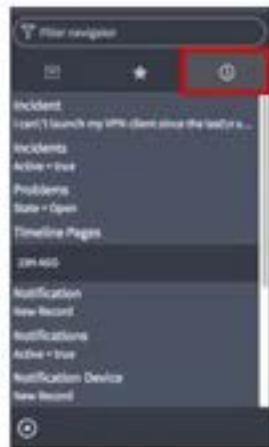


### Your History

Your History provides a scrolling view of recent activities including:

- Forms and lists you accessed
- Homepages you visited

Simply click on any recent activity to open in your Content Frame



**Favorites:** Access the favorites menu to see all your favorites in one place. Favorites include application menus and modules which you may wish to access quickly and often.

**Your History:** The Application Navigator contains a scrolling list of your recent history within ServiceNow. For example, **Your History** will display forms you were filling out or lists you were searching on. Simply click on an item to open any recent activity in your content frame. Some content types are not tracked, including UI pages and other non-standard interfaces.

## Mobile Access

service<sup>now</sup>



In addition to accessing your ServiceNow instance from a laptop or desktop computer, ServiceNow supports the following technologies:

**Smartphone:** The smartphone interface supports many of the features found in the standard desktop/laptop browser interface, including lists, forms, favorite/shortcut management, and filtering. There are no special configurations needed for the iPhone or Android phones; the smartphone interface uses familiar, industry-standard techniques for performing most actions.

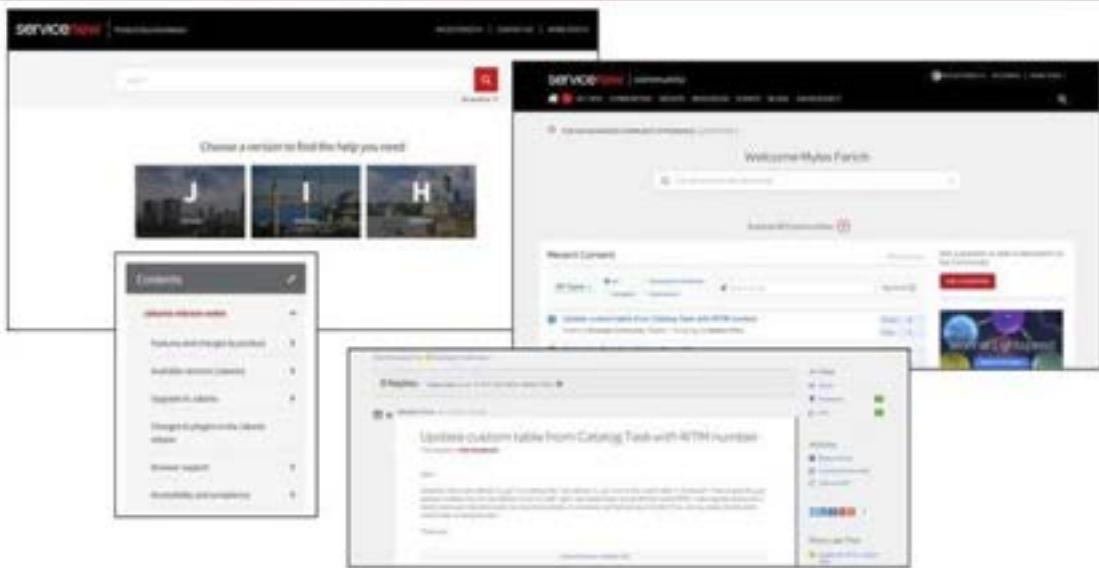
**Tablet:** The ServiceNow instance automatically detects the tablet and redirects to the desktop interface.

**Apple Watch:** Features include: notifications, favorites, record monitoring, chat messaging, dashboard charts, and record interaction via canned responses and voice to text (Siri).

Depending on how you access ServiceNow, the user interface and features may vary.

## Documentation: Docs and Community

servicenow



[docs.servicenow.com](http://docs.servicenow.com) is the official documentation resource for ServiceNow, with content produced by ServiceNow. From features to functionality, and even release notes, this resource should have all of the information needed to get the most out of the platform.

[community.servicenow.com](http://community.servicenow.com) is similar to the Docs website, in that it provides useful information about the ServiceNow platform. However, where Community really excels is by bringing together actual ServiceNow users to collaborate, share, and produce ideas, content, and even answers to questions you may have!

This is a great resource to learn from users with real-life experience on the platform!

## Section Summary

- What is ServiceNow?
- Cloud Infrastructure
- Multi-Instance Software Stack
- Key Platform UI Components
- Mobile Access
- Product Documentation



### Lab 1.1 – ServiceNow Overview:

- Log on to your training instance
- Use the application navigator and its filter to access different areas of ServiceNow
- Impersonate a user from the user menu
- Add My Work and My Groups Work modules to Favorites for easy access later
- Download the ServiceNow mobile app and log on to your instance as the Service Desk user

# Lab Topics

- Log on to your training instance
- Use the Application Navigator and its filter to access different areas of ServiceNow
- Impersonate a user from the User menu
- Add My Work and My Groups Work modules to Favorites for easy access later
- Download the ServiceNow mobile app and log on to your instance as the Service Desk user

This course builds on a scenario where you work for a division of a fictitious electronics company called Cloud Dimensions.

## Lab 1.1 ServiceNow Overview

Upon the reveal of their Infinity product, a portable holographic projector, you will support a team of department Subject Matter Experts (SMEs) with the implementation of ServiceNow.

ServiceNow will be used for tracking Infinity inventory, order fulfillment, and customer support, among other things.

You will be required to impersonate various user personas – representing Cloud Dimension employees – throughout this course's labs.

**NOTE:** Screen shots are often cropped so what you see in the Participant Guide may not match exactly what you see in your instance.

## Log on to Your Training Instance

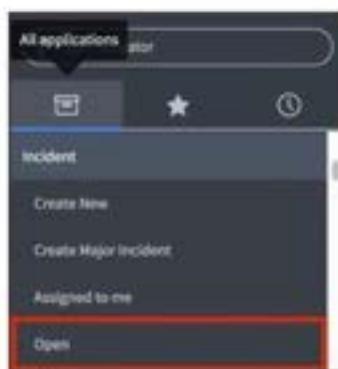
**NOTE:** Your instructor will provide you with your own instance URL.

1. Navigate to **your assigned ServiceNow Lab Instance** in the web browser of your choice.
2. Log on using the **System Administrator (admin)** credentials provided by your instructor.

## Use the Filter Navigator

1. Set the Application Navigator view to display all applications in an expanded view (double-click the **All applications** navigator icon to expand/collapse all), then locate the **Incident** application to view the 9 incident modules.
2. **Incident > Open.**

**NOTE:** The **Application Menu > Module Name** formatting indicates the navigation path to use in the expanded Application Navigator. This shorthand will be used in the lab instructions going forward. For this step, select the **Incident Open** module:

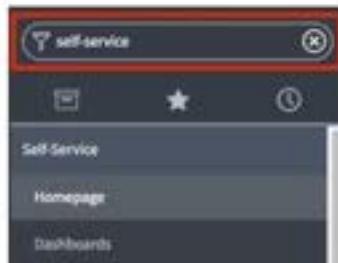


Notice how the user interface changed in the Content Frame from the System Administrator homepage to a list of open incident records.

3. **Incident > Create New.**

**NOTE:** Notice how the user interface has changed in the Content Frame from a list of incident records to an individual incident record/form.

4. From the Application Navigator, use the **Filter navigator** to filter the list of application menus and modules by typing **self-service** into the Filter navigator:



**NOTE:** A single application menu, **Self-Service**, appears with many modules. Scroll down to see all of the modules under the Self-Service application.

5. **Self-Service > Service Catalog.**

**NOTE:** Notice yet another user interface type displayed in the Content Frame.

6. From the Application Navigator, type the keyword **service** into the Filter navigator.

**NOTE:** Scroll to see all of the applications and modules that contain the text "service" display.

## Set Module Favorites

1. Open the User menu on the Banner Frame, then select **Impersonate User**:



2. Impersonate the Service Desk user **Kevin Edd** by typing their name into the **Search for user** field.

**NOTE:** After selecting their name from the drop-down list, ServiceNow should reload and you are now impersonating Kevin Edd:

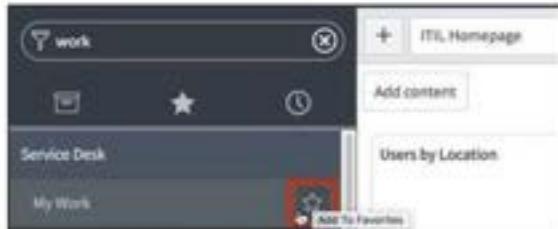


Some modules will be set as favorites under Kevin's account with the intention that they are used as shortcuts in later labs, while impersonating Kevin Edd.

3. Filter the Application Navigator using the keyword **work**.

4. **Service Desk > My Work.**

5. Hover over the **My Work** module, then add the My Work module as a favorite by selecting the **Add to Favorites** icon (star) to the right of the module name:



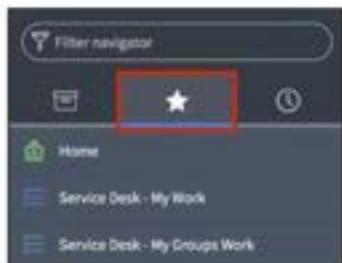
**NOTE:** The star will appear in gray.

6. Repeat this step for the **My Groups Work** module.

**NOTE:** In addition to application menus and modules, the Filter navigator will also display Favorites based on keywords:



7. Clear the Filter navigator keyword by selecting the X to the right of the Filter navigator.
8. Next, navigate to the **Favorites** tab of the Application Navigator to see the module favorites you have created:



9. On the bottom-right of the Application Navigator, select the **Edit Favorites** icon (pencil):



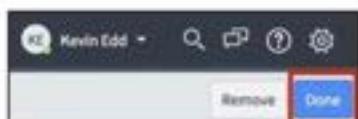
**NOTE:** The **Set up your favorites** screen displays in the Content Frame. A favorite can be customized to have any name, color, and icon.

10. Select the **Service Desk – My Work** favorite in the Application Navigator.

11. Select any color and icon for the **Service Desk – My Work** favorite.

12. Repeat steps 10 and 11 for the **Service Desk – My Groups Work** favorite.

13. Click the **Done** button:



14. Minimize (collapse) the Application Navigator by selecting the **Minimize Navigator** icon (circled arrow) at the bottom of the Navigator:

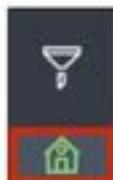


15. From the minimized Application Navigator, notice that the two favorites appear in the color and icon you have selected:



**NOTE:** Your color and icon choices may vary from what is shown here for demonstration purposes.

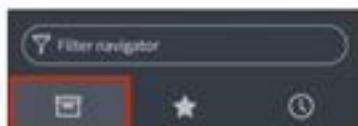
16. Navigate to the homepage by selecting the **Home** favorite displayed on the minimized Navigator:



17. Select **Maximize Navigator** at the bottom of the minimized Navigator:



18. Next, select **All applications**:



## Access ServiceNow from a Mobile Device

Now let us download the ServiceNow app to see what the experience looks like. Accessing ServiceNow from a mobile device is a great way to be updated on record activity, receive chat messages from colleagues, as well as respond to actionable items such as approvals.

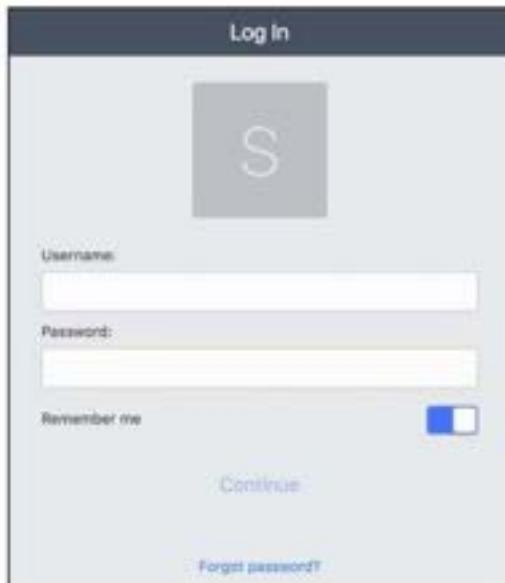
This portion of the lab has three sets of instructions: emulation, iPhone, and Android devices. Please locate and complete the set of instructions relevant to you.

Depending on your company policies or if you do not have an iPhone or Android device, please follow the desktop emulation steps.

### Emulation: Accessing the Desktop Emulator

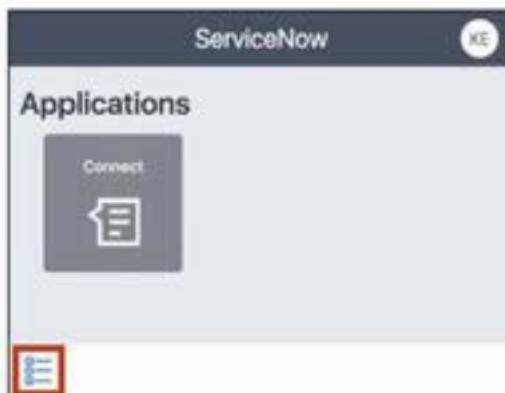
1. Log out of your instance.
2. From your browser, add the **/\$m.do** prefix to the end of the instance URL:  
[https://instance-###.lab.service-now.com/\\$m.do](https://instance-###.lab.service-now.com/$m.do)
3. Press **Enter** on your keyboard.

You should be brought to the mobile interface through emulation:



**TIP:** Resize your browser window to best emulate the size of a phone screen.

4. Enter Kevin Edd's credentials: **kevin.edd / eddpass**
5. Click **Continue** to be brought to the mobile home screen.
6. Open the **Navigator**:



7. Select the **Favorites** icon:



8. The favorites you have created during this lab appear:



**NOTE:** To return to the desktop interface and prepare for the next lab, add the **logout.do** suffix to the end of your instance URL, after **.com/**, then press enter.

### iPhone: Downloading the App

1. Open the **App Store** on your iPhone.
2. Click **Search** from the tray on the bottom.
3. Type **ServiceNow** into the search field at the top, then click **Search** from the keyboard.
4. Click **Get** to the right side of the screen, then click **Install**.
5. If asked, enter your Apple ID password.
6. Once the app has downloaded, open it and then enter **your assigned ServiceNow Lab Instance**.
7. Click **Continue**.
8. Enter Kevin Edd's credentials: **kevin.edd / eddpass**

9. Click **Continue** to be brought to the mobile home screen:

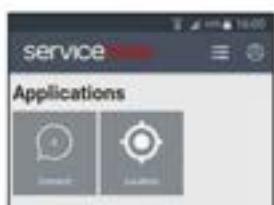


10. Swipe right on the iPhone screen to quickly access the favorites list, where the favorites you have created during this lab appear:

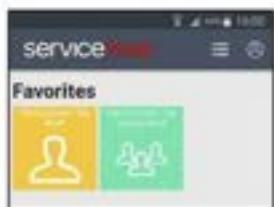


### **Android: Downloading the App**

1. Open the **Play Store** on your Android phone.
2. Type **ServiceNow** into the Google Play search field at the top, then select the keyword from the results list.
3. Click on the **ServiceNow** app icon, then click **Install**.
4. Once the app has downloaded, open it and then enter **your assigned ServiceNow Lab Instance**.
5. Click **Continue**.
6. Enter Kevin Edd's credentials: **kevin.edd / eddpass**
7. Click **Continue** to be brought to the mobile home screen:



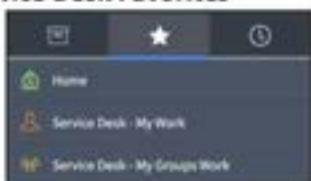
8. Swipe right on the Android screen to quickly access the favorites list, where the favorites you have created during this lab appear:



## LAB VERIFICATION

The Lab Verification shows you screen shots of what you should have created. Sometimes the Lab Verifications have already been shown in earlier steps – as is the case here.

**Service Desk Favorites**

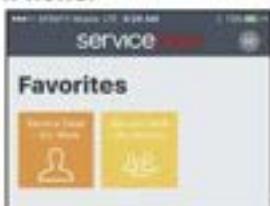


**ServiceNow Mobile App**

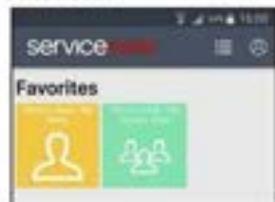
**Desktop Emulator:**



**iPhone:**



**Android:**



*Congratulations, you have completed the ServiceNow Overview lab!*

## Module 1

service*now*

### 1.2 Table Lists

- What is a List?
  - Anatomy
  - Views
  - Controls
- Layout Configuration
- Personalization
- List Editing
- Finding Information: ServiceNow Search
- Tags
- Filters

## What is a List?

servicenow

A list displays a set of records from a table

	Number	Short description	Caller	Priority	State	Category	Assignment group
<input type="checkbox"/>	INC0000003	Wireless access is down in my area	Joe.Employee	1 - Critical	In Progress	Network	Network
<input type="checkbox"/>	INC0000007	Need access to sales DB for the Web	Joe.Employee	1 - Critical	On Hold	Database	
<input type="checkbox"/>	INC0000015	I can't launch my VPN client since the last software update	fred.Luddy	1 - Critical			

Lists and forms are the most common ways to interact with data. A list displays a set of records from a table. You can filter and customize lists to display the information you need.

**NOTE:** You may encounter two different versions of list functionality referred to as List v2 and List v3. List v3 is enabled by a ServiceNow plugin and offers additional functionality such as displaying information in a split format.

## List: Anatomy

servicenow

### 1. Title Bar

Displays the list title and, in some cases, the name of the specific list view

### 2. List Filters/Breadcrumbs

Offers a quick form of filter navigation

### 3. Column Headings

Displays column (field) names and provide some list controls

### 4. Column Header Search

Provides a search within a specific column

### 5. Fields

Displays data; you can right-click a field to access more actions

Number	Short Description	Color	Priority	State	Category
INC000002	Unable to get to network file shares	Red	1-Critical	On Hold	Network
INC000003	Need access to server-28 for the user	Yellow	1-Critical	In Progress	Network
INC000007	Need access to server-28 for the user	Yellow	1-Critical	In Progress	Database
INC000012	I can't launch my VM client since the last software update	Red	1-Critical	In Progress	Software
INC000014	Rain is leaking on my DNS Server	Yellow	1-Critical	In Progress	Hardware
INC000017	How do I create a sub-tenant?	Yellow	1-Critical	On Hold	Inquiry/Help
INC000018	Sales forecast spreadsheet in Risk ONLY	Yellow	1-Critical	In Progress	

Although lists display data captured in different tables, their interface remains consistent with common features.

## List: Views

servicenow

A view is a version of a customized list or form which defines the layout order and what fields appear on the list or form.

For list views, the same number of records for that particular table display – different fields may be visible and display in a different order.



Default view



Mobile view

**Views:** Views enable users to quickly display the same list or form in multiple ways. System administrators can create views for lists or forms. For example, you can create and use different views in Incident for an ESS user, an ITIL user, and a mobile user.

To switch between the different views of columns on a list (as shown here), open the List Control Menu then select **View**. Then, select the name of the desired view.

The view name appears in brackets beside the table list title and form record type when a view other than the Default view is selected.

**NOTE:** Switching views on a form will attempt to save all changes made to the record. You will receive a message asking if you want to save or discard all changes made to the record, before the form reloads and displays the selected view.

**Sort Controls:** A list that is displayed to a user for the first time will be sorted by one of the following:

- The **order** field, if one is present in the table
- The **number** field, if one is present in the table
- The **name** field, if one is present in the table
- The field specified as the display field for the table

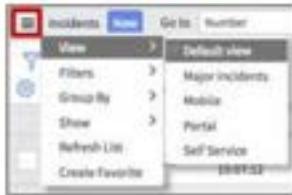
## List: Controls

servicenow

**Control menus**, also sometimes called Additional Actions or context menus, provide different levels of controls for a given list

Control menus can be accessed by clicking the list menu icon ( $\equiv$ ) or by right-clicking the list header and column headers respectively, and only for the **Record Control Menu**, right-clicking in a row's cell

### List Control Menu



### Column Control Menu



### Record Control Menu



List Control (or context) menus, also sometimes called Additional Actions, can be accessed from lists, columns, or on records by using right-click menus which provide different levels of controls:

- **List Control Menu:** You can click the list control menu icon next to the title of the list (Incidents in this example) to access options related to viewing and filtering the entire incidents list.
- **Column Control Menu:** Click the column control menu icon in the desired column header to display actions related to that column, such as creating quick reports, configuring the list, and exporting data.
- **Record Control Menu:** You can right-click in a row's cell to see a menu with actions related to the values in that cell, such as filtering options, assigning tags, and more.

## List: Layout Configuration

servicenow

Configuring a list allows you to show or hide fields from a view, as well as change the list column order

Click the column control menu icon and select **Configure > List Layout**



If you have the **admin** or **personalize\_list** role, you can add or remove columns (fields) from a list or change the order in which the columns appear in the list, for all users.

To do so, navigate to the list, then open the **Column Control Menu**, then select **Configure**, finally, select **List Layout**.

The slushbucket opens and has two sections: the available items on the left, and the selected items on the right. Items from the available section can be added to the list and items from the selected section can be removed from the list.

Once items are in the selected section, there are controls to adjust their order (up or down) on the list.

**List: Personalization**

Personalizing a list allows you to show or hide fields on a view, as well as change the list column order

Unlike configuring a list, personalizing will not affect what others see on their lists

To personalize a list view, click the **Personalize List** icon (⊕) from the list column header

After personalizing a list, this icon will change to look like this (⊕)

**Personalize List Columns** modifies a list for an individual user; it does **not** affect the platform default. List layout changes made using **List Layout** will affect everyone, across the entire instance, except for individuals using personalized layouts set via **Personalize List**.

Through Personalize List Columns, you can do the following:

- **Add Columns:** In the available section, select each column you want to add and press the add icon
- **Remove Columns:** In the selected section, select each column you want to remove and press the remove icon
- **Rearrange Columns:** In the selected section, select the column(s) you want to reorder and use the up or down icons to place the columns in the desired order
- **Reset Column Defaults:** Return the list's columns to the default list's view definition

## List: List Editing

servicenow

The **List Editor** allows you to edit field values in a list without opening the form.

Locate a record with the field value you want to change:

1. Double-click in an empty area of the field
2. Enter the appropriate value(s)
3. Save the record by clicking the save (✓) icon  
– Clicking the cancel (✗) icon or pressing the Escape key retains the original value



Users can edit data in lists using various methods but certain field types cannot be edited.

You can edit multiple records at the same time using the list editor. If you want to update a single field on multiple records to have the same value, the list editor is the quickest method.

### Procedure

1. Select the records you want to edit.
2. Open the list editor by double-clicking (or clicking, depending on setup) in an empty area of the field. The number of selected rows that you are editing is indicated. If any rows cannot be edited due to security constraints, that is indicated. Administrators can configure the list editor and by default, list editing is disabled for some tables.
3. Enter the appropriate values and click the save icon.

Quick edit functions may also be used to edit records. Right-click a field and select the appropriate function:

- **Assign to me:** For records that use assignments, places the logged-in user's name into the **Assigned to** field
- **Approve:** For records that use approvals, changes the approval state of the record to **Approved**
- **Reject:** For records that use approvals, changes the approval state of the record to **Rejected**
- **Assign tag:** For records you want to track based on a user-defined label

## Finding Information: ServiceNow Search

service<sup>now</sup>

Use any of the following searches to find information in ServiceNow:

- **Lists:** Find records in a list; search in a specific field (Go to), all fields (Search), or in a specific column
- **Global text search:** Find records in multiple tables from a single search field
- **Knowledge Base:** Find knowledge articles
- **Service Catalog:** Find catalog items
- **Filter Navigator:** Filter the items in the application navigator

### List and Global Wildcard Searches

Wildcard Search Syntax	Does a Search
*mySearchTerm	Contains
!*mySearchTerm	Does not contain
mySearchTerm%	Starts with
%mySearchTerm	Ends with
=mySearchTerm	Equals
!=mySearchTerm	Does not equal
mySearchTerm No leading or trailing Wildcard(s)	Greater than or equal to mySearchTerm when using the list Go to search only

Find information quickly in ServiceNow by using any of the available searches:

- **Wildcards:** Use a symbol to represent zero or more characters
- **Phrase Searches:** Find a phrase with multiple terms
- **Searching Lists:** Control the query for list searches of a specific field
- **Boolean Operators:** Refine searches with operators such as AND and OR
- **Attachment Searches:** Search in files that are attached to Knowledge Article records
- **International Character Sets:** Perform searches with any Unicode characters
- **Punctuation:** Perform searches that contain punctuation

Wildcards use a symbol to represent zero or more characters and are available for searches. You can use various wildcards to refine your search in lists (text searches of all fields), the global text search, and the Knowledge Base. Results with using wildcards may vary depending on the search method used.

Searches are not case sensitive. Use advanced options for more specific queries.

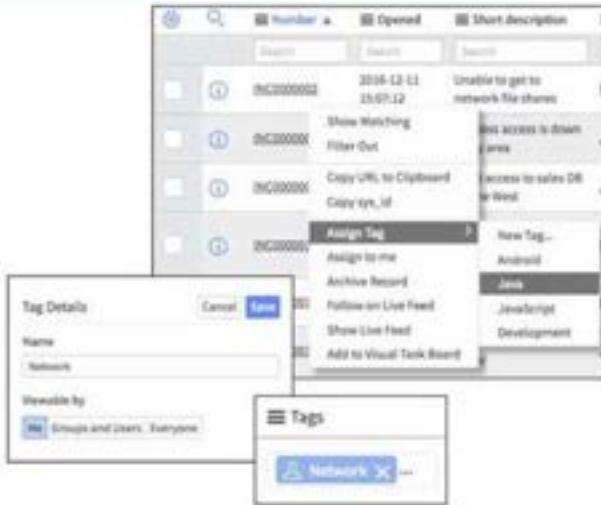
**NOTE:** Zing is the text indexing and search engine that performs all text searches in ServiceNow.

## Tags

service<sup>now</sup>

Create personal **tags** to categorize, flag, and locate your records

- Allow the grouping and organizing of records
- Can be visible to any user (global) or visible only to specific users
- Can be created against any record from a list or from the form



There are a few ways to assign tags to records:

- From the list view using inline field editing
- From a list using the record control menu
- Configuring tags to assign automatically

Use the **Viewable by** field when editing a tag to control how it is shared: visible only to the owner (Me), visible to the owner and specific groups or users (Groups and Users), or visible to everyone (Everyone).

To use the Everyone option under Viewable by, you must have the **admin** or **tags\_admin** role.

**NOTE:** You can edit tags you have created using the **My Tags** or **My Tagged Documents** modules.

## What are Filters and Breadcrumbs?

servicenow

### Filters

A filter is a set of conditions applied to a table to help you find and work with a subset of the data in that table.

### Breadcrumbs

Breadcrumbs offer a quick form of filter navigation and are ordered from left to right.



**Filters** allow you to specify exactly which records you want to see within a selected list. For example, you may start with a list of all incidents but filter those records to view only active incidents assigned to you. Users can apply, create, modify, and save filters.

To view the filter applied to a list, click the funnel icon on the top left of the list. Here, you can add, remove, or edit **filter conditions** and rerun or save your filter.

Click **Run** to see the results of your filter, displayed in the list. To save a filter, click **Save**. A new field will appear where you can name your filter. After naming the filter, click the **Save** button to the right of the name field. The new filter will be available by selecting **Filters** from the List Control Menu.

The filter conditions applied to the list are summarized in the **breadcrumbs**, shown in blue letters across the top of the list. Not only do the breadcrumbs provide an “at-a-glance” view of the filter’s conditions, but you can also modify conditions and add to favorites your filter directly from the breadcrumbs.

Click a breadcrumb to remove all conditions to its right. Clicking the condition separator (>) before a condition to remove only that condition.

Add filter conditions to your favorites menu simply by dragging and dropping the breadcrumbs onto the navigator.

## Filter Conditions and Filter Operators

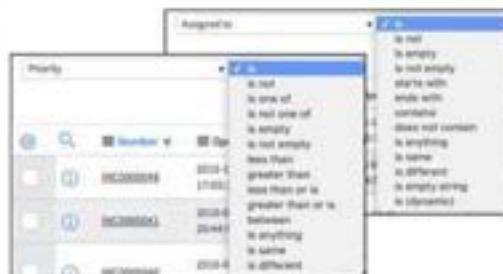
servicenow

### Three components of a filter condition:

1. Field
2. Operator
3. Value

The screenshot shows a search interface with three filter conditions. Condition 1 (Active) has an operator of 'is true'. Condition 2 (Assigned To) has an operator of 'is one of' and a value of 'System Administrator'. Condition 3 (Short Description) has an operator of 'starts with' and a value of 'abc'.

A **filter operator** represents the action you want to take; it is a choice list based on the field type



The three parts of a filter condition are:

1. **Operator:** A **choiceField**: A choice list based on the table and user access rights. The choice list includes fields on related tables by dot-walking.
2. **e list based on the field type.** For example, in the incident table, the greater than operator does not apply to the Active field but it does apply to the Priority field.
3. **Value:** A text entry field or a choice list, depending on the field type. For example, in the incident table, the Active field offers a choice list with the values true, false, and empty, while the Short description field offers a text entry field.

A filter operator can specify conditions including: it is this, it is not this, it is same as, it is different from, etc.

Filter operators will change depending on field data type, for example:

- **Text value:** is, is not, contains, is one of, starts with, ends with
- **Numeric:** is, is not, greater than, less than, greater than or is, less than or is
- **Date:** on, before, after, between, is more than, is less than

## Section Summary

- What is a List?
- Layout Configuration
- Personalization
- List Editing
- Finding Information
- Tags
- Filters

### Lab 1.2

List and Filters



Pages 47 – 56



15 – 20 minutes

### Lab 1.2 – Lists and Filters:

- Open an Update Set
- Create a new Infinity list view on incident for employee incidents related to testing
- Practice filtering data on the incident list and saving a new filter for later use
- Update records using inline editing

# Lab Topics

- Open an Update Set
- Create a new Infinity list view on incident for employee incidents related to testing
- Practice filtering data on the incident list and saving a new filter for later use
- Update records using inline editing

One goal of Cloud Dimensions with using ServiceNow is handling Infinity support.

Before the product is launched, however, Cloud Dimensions employees are actively testing Infinity devices.

Winnie Reich – manager of the Service Desk – has requested help from the Cloud Dimensions System Administrator in creating a new Infinity view on the incident table.

This view will be configured to include the necessary fields for supporting Infinity, for both internal and external users alike.

Winnie has also asked her direct report, Kevin Edd, to create and share a list filter that will filter active incidents and display only those submitted by Infinity employee testers.

## Lab 1.2 Lists and Filters

### Open an Update Set

Before starting the lab, create an Update Set. Update Sets are used to capture configuration changes made to the platform and are explored later in class. Create an Update Set now and follow up with it at a relevant time.

1. Impersonate the **System Administrator** using the user menu.

**NOTE:** If you have logged out of your instance, use the **System Administrator (admin) credentials** provided by your instructor

2. **System Update Sets > Local Update Sets.**
3. From the list of local Update Sets, click the **New** button.

- Fill out the form as follows:

Name: **ServiceNow Fundamentals**

State: **In progress** (auto-fills)

Description: **Includes work completed in the ServiceNow Fundamentals course.**

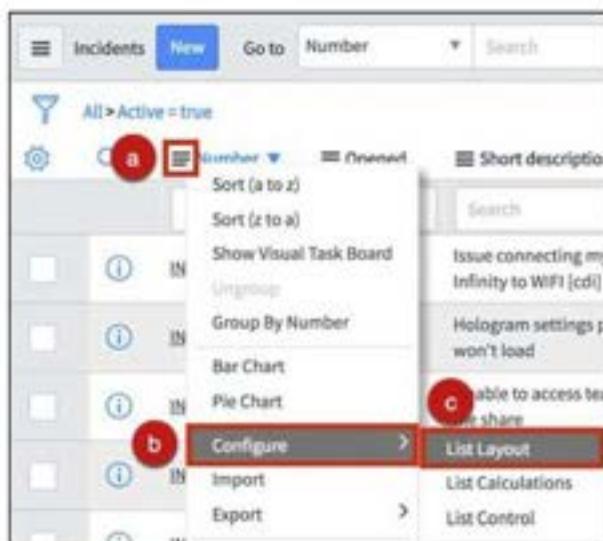
- Click **Submit and Make Current**.

**NOTE:** A message appears at the top indicating this Update Set has been set to the current Update Set. This means changes made to the platform will be captured in this Update Set moving forward.

## Create the Infinity List View

The System Administrator user has the appropriate permissions for creating a new list view on incident – we will assume they have already received the requirements from Winnie Reich.

- Incident > Open.**
- From the list column header, open the “slushbucket” to create a new list view:
  - Open the **Column Control Menu**
  - Select **Configure**
  - Select **List Layout**



**NOTE:** Selecting any field will work, but in this example Number was used.

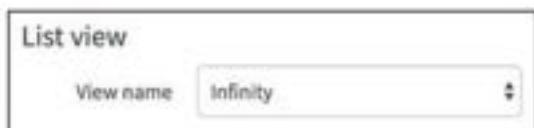
3. Beneath the Available and Selected buckets, open the **View name** drop-down menu under the List view section.
4. Select **New...** at the bottom of the list:



**NOTE:** Choosing an existing view from this list will allow you to modify it.

5. Enter the View name: **Infinity**.
6. Click **OK**.

Nothing appears to have happened to the page but you should now notice **Infinity** as the selected List view:



7. Working with the **Available** and **Selected** buckets, use the **Add** and **Remove** buttons (">" and "<" icons, respectfully) to create the Infinity list view with the following fields:

**Number**  
**Priority**  
**State**  
**Caller**  
**Category**  
**Subcategory**  
**Short description**  
**Assignment group**  
**Assigned to**  
**Tags**  
**Updated**

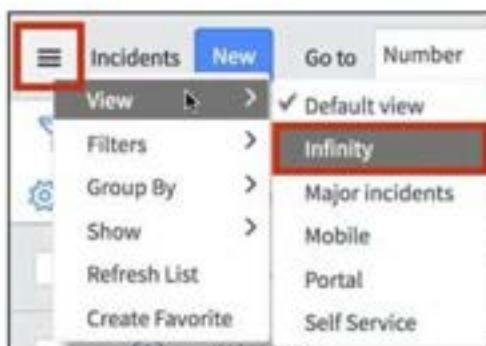
**NOTE:** Ensure the fields are listed in this same order under **Selected**. Use the **Move up** and **Move down** arrows, on the right of Selected, to set the correct order.

**TIP:** You may select multiple fields under Available or Selected, then add or remove them with one click.

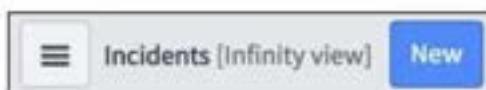
8. Once the fields have been added to the Selected bucket, click **Save**.
9. Impersonate **Kevin Edd** to confirm the view is available for the Service Desk group.
10. As Kevin Edd, navigate to **Incident > Open**.

**TIP:** Add the **Incident > Open** module to favorites, as it will be accessed by Kevin frequently.

11. Open the **List Control Menu** and select **View**, then finally select **Infinity**:



**NOTE:** The list view name now appears at the top of the list in square brackets:

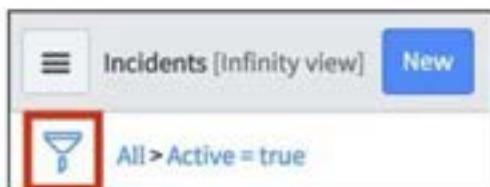


12. Confirm the fields appear in order, from left to right, as listed in step 7 above.

### Apply and Save a Filter

Filters allow users to locate specific data quickly, and filters are also reusable. Kevin Edd will apply a filter that displays Cloud Dimensions Infinity incidents related to employee testing, then save the filter to share with his team for future use.

1. Open the filter condition builder by selecting the Show / hide filter icon (funnel):



2. Add the following AND condition:

#### Tags | EIT

**NOTE:** This will search for all active incident records with EIT as one of its tags. The EIT tag is something Cloud Dimensions employees have created to help distinguish testing incidents from customer incidents – it stands for **Employee Infinity Testing**.

The filter should look like this:

3. Click Run to apply the filter.

There should be two incident records returned.

4. Open the filter condition builder again to save the filter for later use.

5. Click Save...

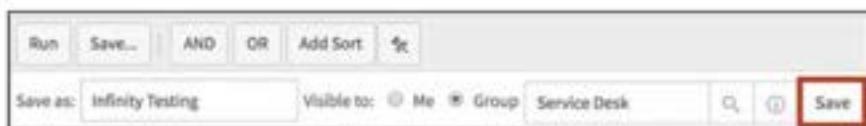
6. Enter Infinity Testing into the Save as field.

7. Next, select **Group for Visible to**.

**NOTE:** The ability to select a group to share a filter with is provided by additional user permissions. For this exercise, the Service Desk group has been given the **filter\_group** role.

8. Input **Service Desk** into the group reference field to share this filter with its members.

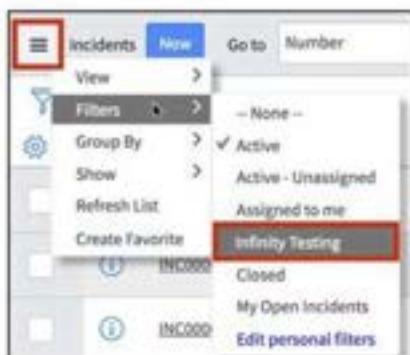
9. Click **Save**:



10. Impersonate **Megan Burke**, another member of the Service Desk group, to confirm the filter is now available for the Service Desk group.

11. **Incident > Open**.

12. Open the **List Control Menu** and select **Filters**, then finally select **Infinity Testing**:



13. The filtered incident list, containing two records, appears:

Number	Opened	Short description	Caller
INC0000052	2017-08-10 09:25:17	password reset request	Megan Burke
INC0000023	2017-08-10 09:24:23	infinity showing an error - employee testing	Ted Keppel

## Locate a Missing Incident

Winnie Reich has emailed Kevin Edd to report an incident submitted by another employee that did not follow the current EIT tagging convention:

**INC0000061** was submitted by Alissa Mountjoy and will need to be updated to include the EIT tag as the incident reports an error found with the Infinity holographic settings page.

1. Impersonate **Kevin Edd**.
2. From the open incidents list, locate and open Alissa's record.

**HINT:** Using the information provided by Winnie, locate the record by using list column header searches, filter conditions, or a global search for **INC0000061**.

3. With the record form displayed, open **More options** from the form header:



4. Click **Add Tag**.
5. Type **EIT** into the **Add tag...** field.
6. Press **Enter** on your keyboard to add the tag to the incident record:



7. **Update**.
8. Apply the **Infinity Testing** filter to the incident list to confirm all three Infinity testing records display:

Number	Priority	State	Caller	Category	Subcategory	Short description
INC0000061	5 - Planning	New	Alissa Mountjoy	Network		Infinity holographic settings page will not display
INC0000042	5 - Planning	New	Megan Burke	Database		password reset request
INC0000023	5 - Planning	New	Ted Koppel	Inquiry / Help		Infinity showing an error - employee testing

## Update Infinity Incident Records

Now that all Infinity incident records are accounted for, Winnie Reich has asked Kevin Edd to ensure all records' categories are accurate based on the issue reported and described.

Use the inline editor to update a record's category value right from the list.

1. Find **INC0000042** and double-click on the category, **Database**:

INC0000042	S - Planning	New	Megan Burke	Database	password reset request
------------	--------------	-----	-------------	----------	------------------------

2. Use the Category drop-down to select **Inquiry / Help**.
3. Click the **Save** icon (green checkmark) to update the record:

Category:	<input checked="" type="checkbox"/> Inquiry / Help	<input type="checkbox"/>
Subcategory:	-- None --	

INC0000042	S - Planning	New	Megan Burke	Inquiry / Help	password reset request
------------	--------------	-----	-------------	----------------	------------------------

4. Select multiple records to update field values with one set of steps:

- a) Press **Shift** and click the **Network** category for INC0000061
- b) Hold **Shift + Ctrl** (**Shift + Command** on Mac) and click the **Inquiry / Help** category for INC0000023

≡ Number ▾	≡ Priority	≡ State	≡ Caller	≡ Category
INC0000061	S - Planning	New	Alissa Mount	Network
INC0000042	S - Planning	New	Megan Burke	Inquiry / Help
INC0000023	S - Planning	New	Ted K	Inquiry / Help

- Double-click on the **Inquiry / Help** category value for INC0000023 to open the **Category** drop-down.

Notice it indicates two records will be updated:

Category:  
Inquiry / Help ✓  
Subcategory:  
-- None -- ▾  
2 rows will be updated

- Use the **Category** drop-down to select **Software**.
- Save** to update both records.

- Your Infinity incident list should look like the following:

Number	Priority	State	Caller	Category
INC0000061	5 - Planning	New	Alissa Mountjoy	Software
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help
INC0000023	5 - Planning	New	Ted Keppel	Software

## LAB VERIFICATION

### Infinity Incident List View

List view

View name: Infinity

Selected

- Number
- Priority
- State
- Caller
- Category
- Subcategory
- Short description
- Assignment group
- Assigned to
- Tags
- Updated

### Infinity Testing List Filter

All > Active = true > Tags has list

Run Save... AND OR Add Sort % X

All of these conditions must be met

Active	is	true	AND	OR	X
Tags	is		O	AND	OR

### Updated Infinity Incident Records

Number	Priority	State	Caller	Category
INC0000061	5 - Planning	New	Alissa Mountjoy	Software
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help
INC0000023	5 - Planning	New	Ted Keppel	Software

*Congratulations, you have completed the Lists and Filters lab!*

### 1.3 Forms

- What is a Form?
  - Header Icons
  - Field Types
- Reference Fields
- Formatters and Related Lists
- Configuration
- Personalization
- Templates
- Saving Forms

## What is a Form?

service<sup>now</sup>

A form displays fields from one record; users can view and edit the record data

The screenshot shows a ServiceNow form for incident number INC0000052. The form includes the following fields:

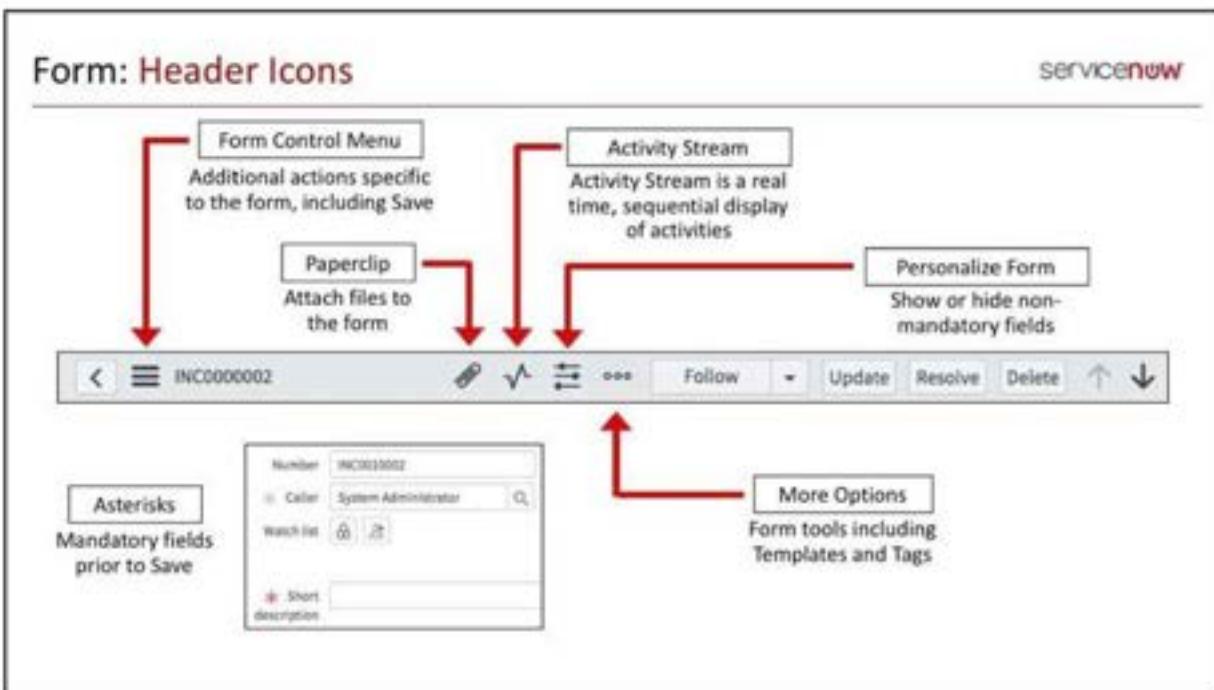
- Number:** INC0000052
- Contact type:** Phone
- State:** In Progress
- Impact:** 1 - High
- Urgency:** 1 - High
- Priority:** 1 - Critical
- Assignment group:** Software
- Assigned:** Seth Argon

Below the form, there are sections for **Notes**, **Related Records**, **Closure Information**, and **Work notes**. A note at the bottom of the form area states: "The SAP HR application is not accessible".

A form displays information from one record in a table. The specific information depends on the type of record displayed. Users can view and edit records in forms. Administrators can configure what appears on forms.

In addition to fields, the form can also contain sections and Related Lists. Related Lists show records in tables that have a relationship to the current record. For example, the User form features Roles and Groups Related Lists. Related Lists do not appear on a form until a record has been saved to the database.

You can load a form directly by searching on a record number in the **Global Text Search** or by clicking in a list.



Each form has different fields, UI actions, and options specific to the application under which it was created.

However, all forms have certain icons and functionalities in common:

- The **Form Control Menu** provides additional options specific to the form. You can find **Save** in the Form Control Menu and use it to save your form while remaining on the page.
- Use the paperclip icon to attach, remove, or rename files on your form.
- **Show Activity Stream** will display a time stamped history of all actions taken within a record.
- You can **Personalize** your form to show or hide important fields. **NOTE:** You cannot hide mandatory fields.
- All fields marked with an asterisk are mandatory and must be filled out prior to saving the form. **NOTE:** The asterisk is red prior to filling out the field and grey once information has been entered
- Click **More options** to tag your form, use templates, send an email, and more.

## Form: Field Types

servicenow

1. String
2. Choice
3. True / False

4. Date / Time

5. Reference

User ID	Name	First Name	Last Name	Email
1	Bernard Lefebvre	Bernard	Lefebvre	bernard.lefebvre@example.com
2	Beth Anglin	Beth	Anglin	beth.anglin@example.com
3	Sue Ruppert	Sue	Ruppert	sue.ruppert@example.com
4	Bud Richman	Bud	Richman	bud.richman@example.com

Reference fields display records from another table

In this example, **Assigned to** references records on the **User** table

Forms include various field types, each with unique attributes.

Some common field types include:

1. **String:** Freely populated using letters, numbers, and special characters. For 254 characters or less, the string field will be a single-line text field. Anything 255 characters or over will appear as a multi-line text box.
2. **Choice:** Drop down list of choices that can be configured.
3. **True/False:** Boolean field that appears as a check box.
4. **Date/Time:** Day and time of day, which can be selected with a calendar widget.
5. **Reference:** Query that displays records from another table.

## Reference Fields

servicenow

**Reference fields** are identified with the **reference lookup icon** (🔍)

The reference lookup icon opens a dialog box for locating a record to reference, presented as a list of the referenced (target) table

If a record is specified in the reference field on the source table, you can hover over the **reference icon** ( ⓘ) to preview the referenced record (on the target table)

**Source Table**  
Incident

Number	Caller	Category
INC0000015	Fred Luddy	Software
INC0000016	Bow Ruggeri	Hardware
INC0000017	Joe Employee	Inquiry / Help

**Target Table**  
User

Name	User ID	Department
Bow Ruggeri	bow.ruggeri	Development
Fred Luddy	fred.luddy	Development
Joe Employee	employee	Sales

A reference field stores a unique system identifier (known as the `sys_id`) of a record on another table which is what establishes the reference relationship. For example, the `Caller` field on the `Incident` table is a reference to a record on the `User` table.

When you define a reference field, the platform creates a relationship between the two tables. Adding a reference field to a form makes the other fields in the referenced table available to the form.

Administrators can create new reference fields and configure several options for reference fields.

**NOTE:** A reference field can refer only to records from one other table. To add a field that can refer to records on any table in the platform, regardless of a shared reference, use the Document ID element type.

Additionally, wildcard searches can be used in reference fields.

## Form: Formatters and Related Lists

service<sup>now</sup>

### Formatter

The screenshot shows a ServiceNow form with a section titled "Activity". Inside this section, there is a "List View" component containing the following data:

Configuration Item	Description	Created Date
SAP Enterprise Resource	Several incidents have been reported by SAP employees. It has been determined that there is an average of 6 SAP incidents per day. We are performing root cause analysis.	2017-02-21 13:58:00
SAP ERP	6 - Low	2017-02-21 13:58:00
SAP Financial Accounting	6 - Low	2017-02-21 13:58:00

A **formatter** is a form element used to display information that is not a field in the record

The activity formatter provides an easy way to track items not saved with a field in the record, for example, journal fields like comments and work notes

### Related List

The screenshot shows a "Related List" for an incident record. The list is titled "Incidents (0) | Problem Tasks". It includes a header with filters for "Incident", "New", "Edit", "Get to", "Number", and "Search". Below the header is a table with the following data:

Number	Opened	Short Description
INC0000031	2017-02-28 12:48:32	Manager can't access SAP Controlling Application
INC0000032	2017-02-28 12:48:40	SAP Financial Accounting application appears to be down

Related lists appear on forms and show records in tables that have relationships to the current record

Examples of **formatters** in the base platform include:

- **Activity formatter:** Displays the list of activities, or history, on a task form
- **Process flow formatter:** Displays the different stages in a linear process flow across the top of a record
- **Parent breadcrumbs formatter:** Provides breadcrumbs to show the parent or parents of the current task
- **Approval summarizer formatter:** Displays dynamic summary information about the request being approved
- **CI relations formatter:** Displays on the CI form a toolbar for viewing the relationships between the current CI and related CIs

Like any other list, users can view and modify information in **related lists**, as well as add a new record to the database. A default filter that is applied to a related list when a form loads can be created.

Administrators can configure related lists to appear on forms and in hierarchical lists. Related lists do not have a size limit.

If there are many related lists on a form or many records in the related lists, the form may load slowly.

## Form: Configuration and Personalization

servicenow

### Configuration

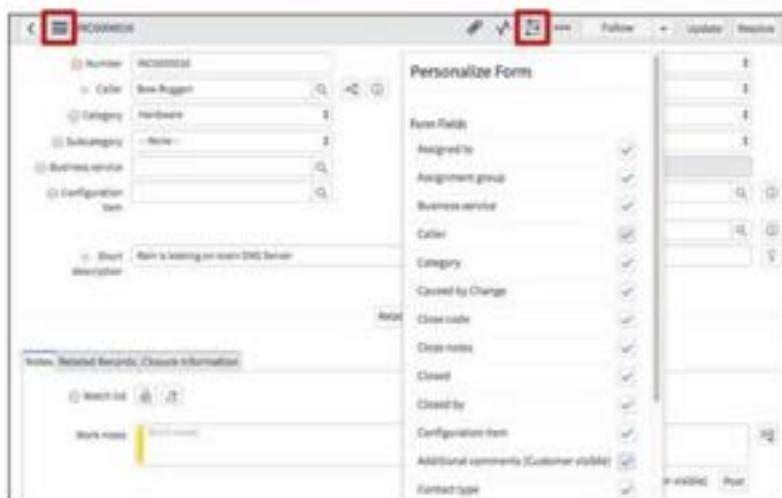
Configuring a form allows you to show or hide fields on a view

Click the control menu icon and select **Configure > Form Layout**

### Personalization

Click **Personalize Form** (≡) from the form header to personalize which fields will appear

Personal form customizations will not affect what others see on their forms



Configure a form to show or hide fields from a view. You can even create new fields on the table that is associated with the form, although this is not best practice.

Using the slushbucket, select the fields and the order in which you want them to appear. Available items that appear in green followed by a plus (+) sign represent reference fields. Accessing these fields on related tables is referred to as dot-walking.

**Warning:** It is not recommended to add the same field to more than one section of a form unless the field displays read-only data. Having two or more instances of an editable field can cause data loss and prevent the proper functioning of UI and data policies.

When the form personalization feature is activated, users can personalize fields to appear on a specific form view according to individual preferences.

In contrast to configuring a form, personalizing a form does not enable users to perform the following actions:

- Change the order of fields on the form
- Add fields that are not configured to appear on the form
- Hide mandatory fields

## Form: Templates

service<sup>now</sup>

**Templates** allow you to populate fields automatically, simplifying the process of submitting new records

Use the template bar at the bottom of the form to manually apply, create, or edit templates

Click the **More options** icon (\*\*\* ) from the form header, then click **Toggle Template Bar**



**Edit Template**

Name: Default ServiceNow Inquiry  
Table: Incident (Incident)  
Active: ✓  
Description: Used for low impact, low urgency inquiries related into the Service Desk  
Template:

Field	Type	Value	Checkmark
Assignment group	Assignment group	Service Desk	X
Category	Category	Priority Help	X
Contact type	Contact type	Phone	X
State	State	In Progress	X
Object	Object	Incident	X
Priority	Priority	Low	X
Urgency	Urgency	Low	X
Custom field	Custom field		

Close Cancel Delete Update

To use a template, populate the most-used fields for a specific table, then save it

To use a template, populate the most-used fields for a specific table, save it as a template, and then make the template accessible to your users. Users can manually apply a template when creating records, or an administrator can define scripts to apply templates automatically. Fields updated by the application of a template will have a checkmark icon next to the field label.

Create templates for the forms that you use frequently, such as incident, problem, and change. There is no limit to the number of templates that you can create or access, but having many templates for each form makes the templates more complex to manage.

**NOTE:** Template creation should be restricted to select groups as it can be used to bypass process, like mandatory fields, UI policies, etc. This is especially important for any record using condition based workflows.

## Form: Saving Forms

service<sup>now</sup>

Save records by using one of the following methods:

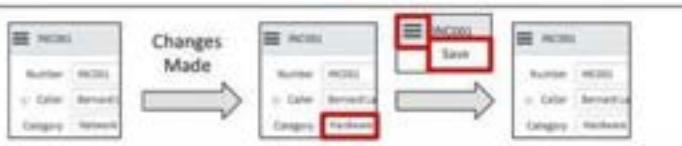
Click **Submit** to save changes on a new form and return to the previously viewed page



Click **Update** to save changes on an existing record and return to the previously viewed page



Select the Form Control menu icon in the header bar, then select **Save** to save changes without leaving the form view



The Insert, and Insert and Stay options are disabled by default for task records such as incidents and change requests but you can configure these options for task records.

**NOTE:** These options are enabled by default on User and CMDB records because they enable bulk entry of similar items.

Once enabled, you can select the Form Control menu icon in the header bar then select **Insert** or **Insert and Stay** to save a new record to the database instead of updating the current item.

There is no "Copy" but **Insert** emulates copy functionality and leaves the form.

**Insert and Stay** does the same but stays on the form.

When a form is saved, all the text in the Work Notes field is recorded to the Activity Log field. Work Notes and Additional Comments are fields that share information with various users associated to certain record types like incident or problem. Additional Comments are visible to all users accessing the record, whereas Work Notes are visible to only users with the itil role. The content in Additional Comments is emailed to the Watch List and Caller, and the content in Work Notes is emailed to the ITIL Watch List and Assigned to user when the form is saved.

**NOTE:** If you make changes to an existing record and then attempt to leave the form (whether using web browser controls such as the 'Back' button, or through the ServiceNow user interface), you will be prompted with a message asking if you are sure you want to leave the record without saving.

## Section Summary

- What is a Form?
- Reference Fields
- Formatters and Related Lists
- Configuration
- Personalization
- Templates
- Saving Forms

### Lab 1.3

Forms and Templates



Pages 67 – 78



20 – 25 minutes

#### Lab 1.3 – Forms and Templates:

- Create and configure a new Infinity form view
- Create a default incident template
- Create incident records for Infinity, using the new template

**Lab Dependency** – This lab requires the completion of Lab 1.2.

# Lab Topics

- Create and configure a new Infinity form view
- Create a default incident template
- Create incident records for Infinity, using the new template

**Lab Dependency** – This lab requires the completion of Lab 1.2.

Internal employee testing of Infinity has proven worthwhile for a number of reasons.

Winnie Reich will lead an initiative to further improve and organize Infinity support by creating a form view on the incident table containing appropriate fields and capable of capturing field values that accurately identify reported issues.

Additionally, she will improve the incident resolution process by requesting the creation of a template for Service Desk agents to use, which will speed up their work by automatically populating common field values on new incident records.

## Lab 1.3 Forms and Templates

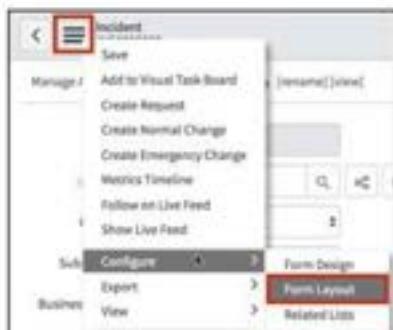
### Create the Infinity Form View

1. Impersonate Winnie Reich.

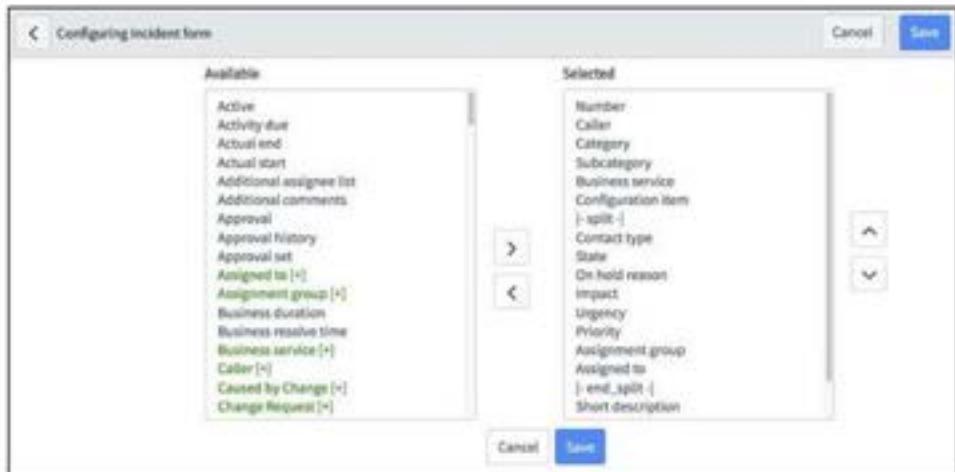
**NOTE:** Winnie is inheriting the **personalize** role from a group she belongs to, which allows her to configure forms and create new views.

2. Incident > Open.
3. Open the record for INC0000061.

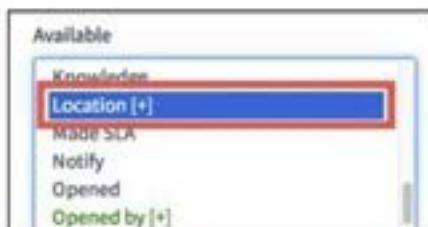
4. Open the **Form Control Menu** and select **Configure**, then finally select **Form Layout**:



The **Configuring Incident form** page displays:



5. Find and highlight the **Location** field under the Available list:



6. Click the **Add** button (>) between the Available and Selected list:



7. Doing this adds the Location field to the bottom of the Selected List.
8. Use the **Move up** button (^) to move the Location field under Configuration item.
9. From the **Configuring incident form** page, click **Save** to return to the incident form.
10. Open the **Form Control Menu** and select **Configure**, but then select **Form Design**.

The **Form Designer** will open in a new tab or window. Go to the page that looks like this:



The **Form Designer** offers an improved experience because of its graphical user interface, making it easier to visualize the form view's end result.

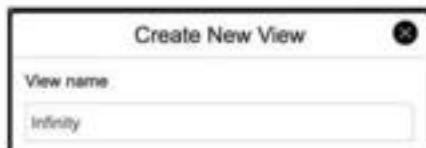
Additionally, there are several configuration options available in this single interface.

At the top left of the page are two drop-down menus in the header; the menu on

the left indicates the table a form view is being designed for, and the menu on the right includes the various views defined for the table.

11. Open the view menu and select **New...** at the bottom of the list.

12. Enter the View name: **Infinity**:



13. Click **OK**.

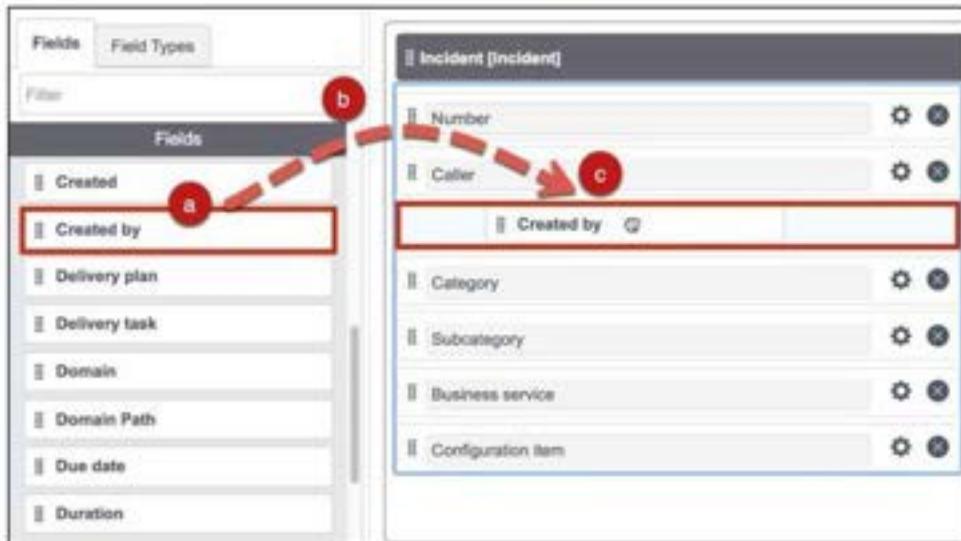
Notice the new view is automatically selected in the view menu on the page's header:



14. On the **Fields** tab of the Field Navigator on the left, scroll down to locate the **Created by** field.

15. Add the **Created by** field to the form view:

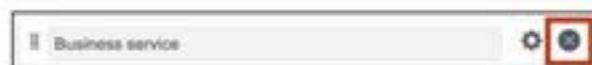
- Click and hold on the **Created by** field
- Drag **Created by** to the form layout, between **Caller** and **Category**
- Release your click to add the field



16. Repeat these steps to add the **Updated** and **Updated by** fields to the form layout, within the Incident section.

**NOTE:** We will be reordering fields later.

17. Find the **Business service** field on the form layout, then click the Remove this field icon (circled x) to remove it from the view:



**NOTE:** Removing a field returns it to the **Fields** tab of the Field Navigator, so it may be re-added if desired.

18. Click and drag the **Contact type** field to be listed beneath the **Number** field:



19. Repeat this step to reorganize the fields in the Incident section to match this layout:



20. From the Field Navigator, click the **Field Types** tab to add a new field to the form layout:



21. Scroll down to find the **True/False** field type, then add the field to the form layout under the **Caller** field:



22. Click the **Edit this field** icon (gear) to configure the field's properties:



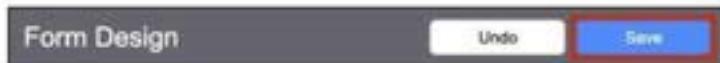
23. Input the following values:

Label: Employee  
Name: u\_employee

**NOTE:** The name features the prefix "u\_" to indicate it is a user-created item. This is a common naming convention used throughout ServiceNow.

24. Close the **Properties** window by clicking the close icon (circled x).

25. Click the **Save** action from the page header to save the form view:



26. Close the **Form Designer** tab/window, and return to the ServiceNow instance.

27. Use the **Form Control Menu** to reload the form (**Reload form**).

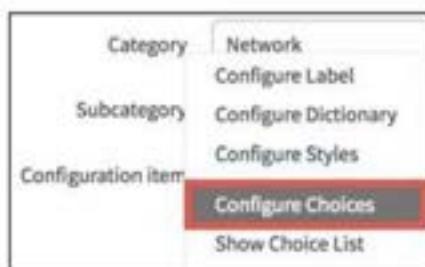
28. Now open the **Form Control Menu** and select **View**, then finally select **Infinity**.

The new **Infinity** form view should load and display as designed!

### Configure the Infinity Form View

Winnie will now configure a couple of fields and their values on the form view, so that the new data can be applied with the template.

1. From **INC0000061**, right-click the **Category** field label.
2. Select **Configure Choices**:



3. On the **Configuring Category Choices** page, input **Infinity** into the **Enter new item** field, located beneath the **Available** bucket:

A screenshot of a form field labeled 'Enter new item:'. The input field contains 'Infinity' and has a red border around it. To the right is a button labeled 'Add'.

4. Click **Add**.
5. Click **Save** to return to the incident record and confirm the **Infinity** choice appears last on the list for the **Category** field.
6. Select **Infinity** from the **Category** field.
7. Next, right-click the **Subcategory** field label and select **Configure Choices**.

**NOTE:** When prompted about leaving the page, confirm your decision and leave.

8. On the **Configuring Subcategory Choices** page, input **Hardware** into the **Enter new item** field, located beneath the **Available** bucket:

A screenshot of a form field labeled 'Enter new item:'. The input field contains 'Hardware' and has a red border around it. To the right is a button labeled 'Add'.

9. Click **Add**.

10. Add these other subcategory choices:

**Network**  
**Request**  
**Security**  
**Software**

11. Your page should look like the following:



12. Save.

**NOTE:** You may be returned to a list of incidents. If you do, open **INC0000061**.

13. From the **INC0000061** record, select the **Infinity** category and confirm the 5 subcategory choices are available to select:

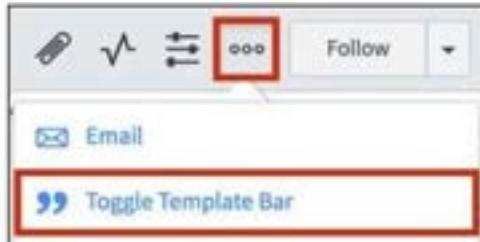


## Create an Incident Template

Adding to the work completed by Winnie, the System Administrator has the correct permissions to create an incident template to be used by Service Desk representatives.

1. Impersonate **System Administrator**.
2. Locate and open **INC0000061** using the strategy of your choice.
3. Apply the **Infinity** view by using the **Form Control** menu.

4. Create a template by first clicking the **More options** form header icon, then selecting **Toggle Template Bar**:



5. Ensure the following fields on the incident have these values:

Employee: [checked]  
Category: Infinity  
State: New  
Impact: 3 – Low  
Urgency: 3 – Low  
Assignment group: Service Desk

6. From the template bar, at the bottom of the form, click the **Create New Template** icon (+):



The template is automatically populated with all of the fields on the incident which have values, including those specific values.

**NOTE:** If any of the fields and their values are missing from the **Create New Template** form, add them using the condition builder drop-down menus.

7. Click the **Delete** icon (X) to the right of any field and value combination under **Template** that does not match the information inputted during step 5:

Template:	Assignment group	▼	Service Desk	<input type="button" value="Q"/>	<input type="button" value="X"/>
	Caller	▼	Fred Luddy	<input type="button" value="Q"/>	<input type="button" value="X"/>

8. Repeat this step for all field and value combinations under **Template** until only the information inputted during step 5 remain.

Your template should look similar to this:

Template	Assignment group	Service Desk
Category	Infinity	infinity
Impact	3 - Low	3 - Low
State	New	New
Employee	true	true
Urgency	3 - Low	3 - Low

9. Scroll back up to the top of the **Create New Template** window, if needed, and change the name to **Default Infinity Employee Incident**.
10. Input **Service Desk** into the **Group** field.

This will make the template available to all members of the Service Desk group.

11. Submit.

## Update and Create Incident Records

With the changes made to the **Category** and **Subcategory** fields, all open employee incident records will need to be updated to include the correct field values.

1. As **Kevin Edd**, use the strategies of your choice to update the following records:

Number	Employee	Category	Subcategory	State	Assignment group
INC0000023	True	Infinity	Software	In Progress	Service Desk
INC0000042	True	Infinity	Request	Resolved	Service Desk
INC0000061	True	Infinity	Software	On Hold	Service Desk

**NOTE:** If required, set the values for the following field:

On hold reason: **Awaiting Caller**

2. Once finished updating the records in step 1, **Incident > Create New**.
3. **Toggle the template bar**, if it is not enabled already.
4. From the template bar, click **Default Infinity Employee Incident**.

- Fill in the rest of the fields as follows:

Contact type: **Walk-in**

Caller: **Buster Wubbel**

Subcategory: **Security**

Short description: **Issue discovered with two step authentication**

Description: **Authentication requirements for logging into Infinity are not working as expected.**

**NOTE:** The **Location** field may have populated automatically because of the **Caller** value inputted.

- Save.

- Open **More options** from the form header.

- Click **Add Tag**.

- Search for and select the **EIT** tag.

- 10. Update.**

- If the active incidents list does not display, navigate to **Incident > Open**.

- Apply the **Infinity Testing** filter.

There are now four total open Infinity employee incident records:

Number	Priority	State	Caller	Category	Subcategory	Short description
INC0010008	S - Planning	New	Buster Wubbel	Infinity	Security	Issue discovered with two step authentication
INC0000061	S - Planning	On Hold	Alissa Mountain	Infinity	Software	Infinity holographic settings page will not display
INC0000042	S - Planning	Resolved	Megan Bucke	Infinity	Request	password reset request
INC0000023	S - Planning	In Progress	Ted Knapel	Infinity	Software	Infinity showing an error - employee testing

**NOTE:** The new incident number may be different in your instance.

## LAB VERIFICATION

### Infinity Incident Form View

The screenshot shows the 'Infinity Incident Form View' window. At the top, there's a header bar with icons for back, forward, search, and buttons for 'Submit' and 'Cancel'. Below the header, the form has several fields: 'Number' (INC0010007), 'State' (New), 'Contact type' (None), 'Impact' (3 - Low), 'Caller' (radio button selected), 'Urgency' (3 - Low), 'Employee' (checkbox), 'Priority' (5 - Planning), 'Location' (dropdown), 'Assignment group' (dropdown), 'Category' (Inquiry / Help), 'Assigned to' (dropdown), 'Subcategory' (None), 'Created by' (dropdown), 'Configuration item' (dropdown), 'Updated' (dropdown), and 'Updated by' (dropdown).

### Default Infinity Employee Incident Template

The screenshot shows the 'Default Infinity Employee Incident Template' window. It contains a single template row with the following fields: 'Assignment group' (Service Desk), 'Category' (Infinity), 'Impact' (3 - Low), 'State' (New), 'Employee' (true), and 'Urgency' (3 - Low).

### Updated Infinity Incident Records

Number	Priority	State	Caller	Category	Subcategory	Short description
INC0010008	5 - Planning	New	Buster Wubbel	Infinity	Security	Issue discovered with two step authentication
INC00000061	5 - Planning	On Hold	Alossa Mountjoy	Infinity	Software	Infinity holographic settings page will not display
INC00000042	5 - Planning	Resolved	Megan Burke	Infinity	Request	password reset request
INC00000023	5 - Planning	In Progress	Ted Kepner	Infinity	Software	Infinity showing an error - employee testing

**Congratulations, you have completed the Forms and Templates lab!**

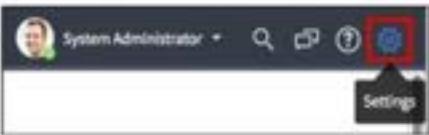
## Module 1

service**now**

### 1.4 Branding

- Settings
- Guided Setup
- Branding Options

## Banner Frame: Settings



Customize your personal Instance using the **Setting** options

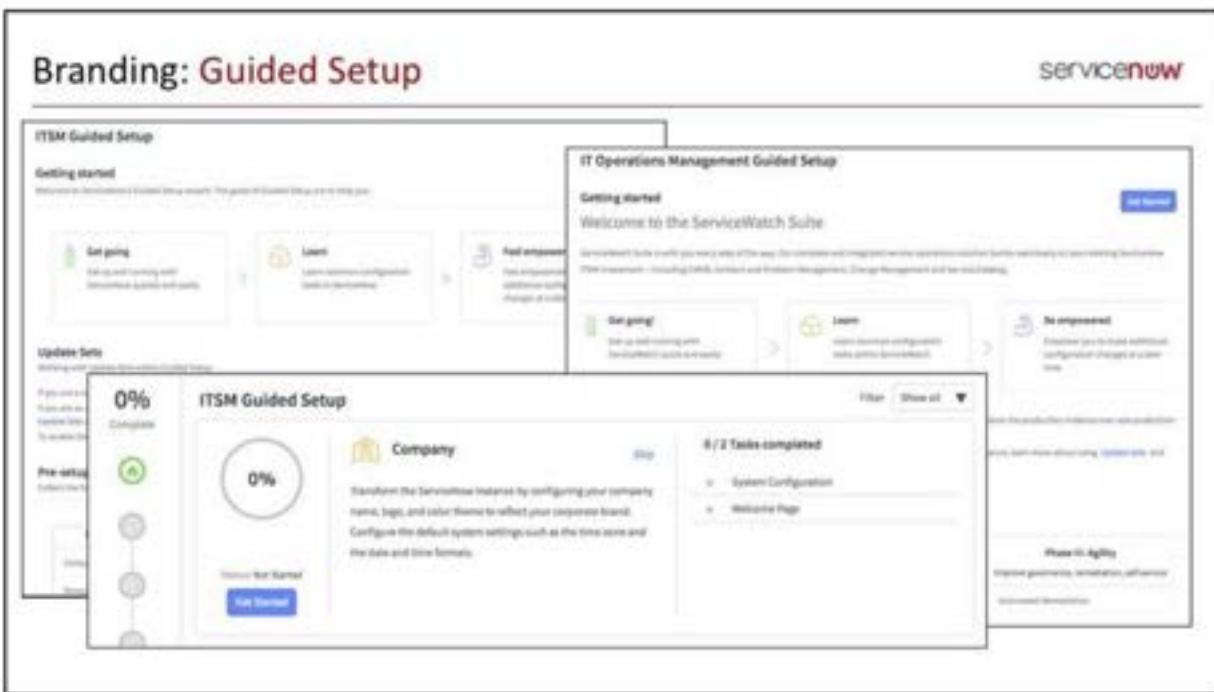
These settings affect only your user account and are retained each time you log in



The Settings icon (gear) in the upper-righthand side of the Banner Frame contains additional settings and options for personalizing your view of the platform.

After selecting the Settings icon, the categories on the left (General, Theme, Lists, Forms, Notifications, and Developer) provide different settings, including:

- **General Tab:** **Compact the user interface** optimizes the UI to display more information in the browser window when this setting is enabled
- **Theme Tab:** Select a theme for the user interface. Select the **System** theme to return to the default theme
- **Lists Tab:** **Wrap longer text in list columns** allows for long strings to wrap in list columns instead of appearing as one long line
- **Forms Tab:** Form sections and related lists appear in tabs when the **Tabbed forms** setting is enabled. Also **Related list loading** is used to determine when Related Lists load on forms.
- **Notifications Tab:** Allows you to enable various notification channels, as well as manage your notification subscriptions
- **Developer Tab:** Settings for ServiceNow Application developers



**Guided Setups** guide you through suggested steps to configure applications on your ServiceNow instance.

Using the **ITSM Guided Setup**, you can perform structured configuration activities that help you configure ITSM applications and can monitor the progress. Each configuration activity in ITSM Guided Setup is designed to simplify the configurations by providing access to contextual embedded help, contextual documentation on the ServiceNow product documentation site, and guided tours (if available for an activity).

The ITSM Guided Setup helps you to configure the common platform settings through the following categories:

- **Company:** Activities under this category help you to configure company name, logo, and color theme to reflect your corporate brand and to configure the default system settings such as the time zone and the date and time formats
- **Connectivity:** Activities under this category help you to configure your ServiceNow instance to support inbound and outbound email notifications and to integrate it with your existing LDAP and Single Sign-On (SSO) solutions
- **People:** In case you do not use LDAP to import data into your ServiceNow instance, activities under this category can help you to import Users, Groups, Group Members, Companies, Departments, and Locations and to assign roles to groups

In addition to the ITSM Guided Setup, there are guided setups for ITOM and HR as well.

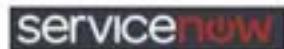
## Branding: Options

servicenow

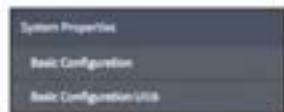
### Browser Tab

Cloud Dimensions ServiceNow

### Banner Image



### Navigation Colors, System Properties



Many branding activities are grouped together in **System Properties > Basic Configuration UI16**.

Customization and Branding options include:

- Banner image, text and colors
- Navigator background and text colors

You can also customize:

- Browser tab title
- Color: Use the built-in color pickers to dynamically pick and preview branding options
- System date/time formatting

## Section Summary

- Settings
- Guided Setup
- Branding Options

### Lab 1.4

Branding



Pages 84 – 88



10 minutes

### Lab 1.4 – Branding:

- Use the Guided Setup to apply branding to the instance
- Create a custom welcome page
- Apply settings

# Lab Topics

- Use the Guided Setup to apply branding to the instance
- Create a custom welcome page
- Apply settings

Cloud Dimensions recognizes the importance of aligning ServiceNow's branding with the rest of the organization so that users automatically feel familiar, seeing it as a trusted platform.

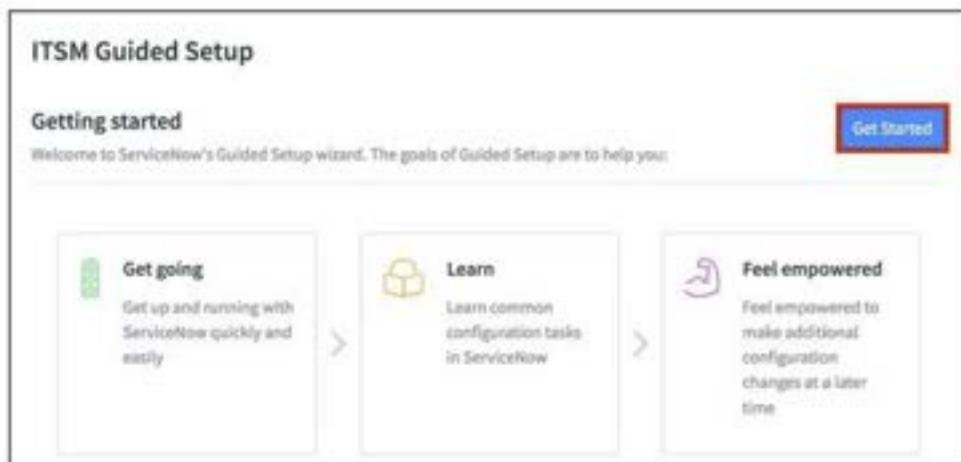
**NOTE:** You will need the `cd-logo.png` file to complete a step in this lab.

## Lab 1.4 Branding

### Apply Branding with the ITSM Guided Setup

Begin the lab by impersonating the System Administrator. This user has the appropriate permissions for defining platform-wide properties. We will work with the ITSM Guided Setup to get started on company branding.

1. Impersonate the **System Administrator**.
2. **Guided Setup > ITSM Guided Setup**.
3. After the Guided Setup page loads, click the **Get Started** button:



4. Next, click **Get Started** from the **Company** section:

The screenshot shows the 'ITSM Guided Setup' interface. On the left, there's a circular progress bar with '0%' and the text 'Status: Not Started'. Below it is a red 'Get Started' button. To the right, there's a section titled 'Company' with a yellow icon. It contains the following text:  
Transform the ServiceNow instance by configuring your company name, logo, and color theme to reflect your corporate brand.  
Configure the default system settings such as the time zone and the date and time formats.

5. The first task to complete is **System Configuration**. Read the summary text, for what to expect, then click the **Configure** button:

The screenshot shows the 'System Configuration' task details. At the top, there are buttons for 'Skip', 'Add Notes', 'Mark as Complete', and a large red 'Configure' button. Below these buttons, the text 'Not started yet' is displayed. The main content area contains two paragraphs:  
Configure default settings for the time zone and the date and time formats of the system. Upload your logo to appear in the banner, and customize the banner and browser tab text.  
Optionally, personalize the background, text, and separator colors of the instance. To do this, obtain the approved brand color names, RGB, or hex values from your marketing group.

6. Fill out the fields as follows:

Page header caption: **Cloud Dimensions**

Browser tab title: **Cloud Dimensions ServiceNow**

Banner image for UI16: **[cd-logo.png]**

Header background color: **#2b3a5a**

Header divider stripe color: **#d84833**

Navigation header/footer and navigation background expanded items: **#486a93**

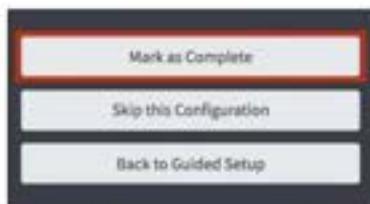
Navigation selected tab background color: **#ffffff**

Background for navigator and sidebars: **#536171**

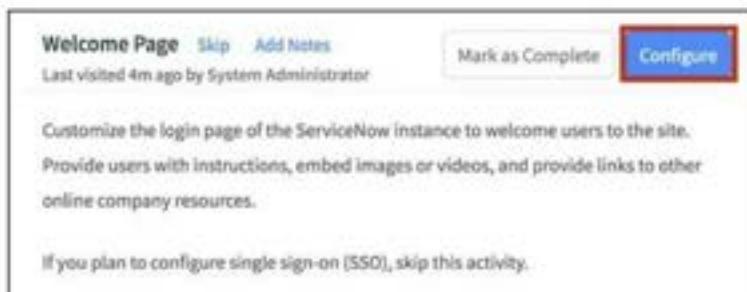
Currently selected Navigation tab icon color for UI16: **#000000**

**NOTE:** As you enter values, especially for colors, you may see these changes in real time.

- When finished, click the **Save** button.
- Refresh your browser to ensure the changes take effect.
- From the **Help** sidebar, click **Mark as Complete** for this Guided Setup task:



- Scroll down on the **ITSM Guided Setup > Company** page, then click the **Configure** button for the **Welcome Page** task:



- Open the record with the Short description `About ${gs.getProperty('glide.product.name', 'ServiceNow')}`.
- Scroll down and update the **Text** field, replacing any existing text, as follows:

**Welcome to the home of Infinity! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.**
- Click **Update**.
- From the **Help** sidebar, click **Mark as Complete**.

The **Company** configuration tasks have been 100% completed:



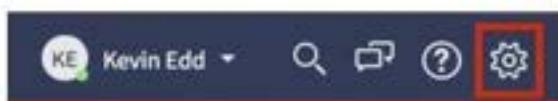
15. Now, Logout to view the changes:

The image shows a ServiceNow login page. It has fields for 'User name' and 'Password', a checked 'Remember me' checkbox, and links for 'Forgot Password?' and 'Login'. Below the login form, there is a message: 'About Cloud Dimensions ServiceNow' and 'Welcome to the home of Infinity! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.'

16. Log back in with the **System Administrator (admin)** credentials.

## Set Settings

1. Impersonate Kevin Edd.
2. Open **Settings** from the Banner Frame:



3. Under **General**, select **Compact the user interface**.

**NOTE:** Toggle this on and off, while watching your instance interface, to see what this does. Why do you think this could be a benefit?

4. Under **Theme**, select anything that is not **System**, so that it is easier to distinguish Kevin Edd from other users.

**NOTE:** For demonstration purposes, as well as clarity of these training materials, screen shots will use the "Cloud Dimensions" **System** theme as defined in this lab.

5. Close out of **Settings**.

## LAB VERIFICATION

### Cloud Dimensions System Theme



### Welcome Page

A screenshot of the Cloud Dimensions ServiceNow Welcome Page. It features a login form with fields for 'User name' and 'Password', a 'Remember me' checkbox, and links for 'Forgot Password?' and 'Login'. Below the form is a section titled 'About Cloud Dimensions ServiceNow' with the text: 'Welcome to the home of Infinity! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.'

**Congratulations, you have completed the Branding lab!**

## Module Recap: User Interface and Navigation

servicenow

**Why** would you use these capabilities?

**When** would you use these capabilities?

**How often** would you use these capabilities?

### ServiceNow Overview

Key Platform UI Components  
Mobile Access  
Product Documentation

### Table Lists

Views  
List Editing  
Tags  
Filters

### Forms

Reference Fields  
Related Lists  
Templates  
Submit vs. Update

### Branding

Settings  
Guided Setups

service**now**



## Module 2

### Users and Tasks

- 2.1 User Administration
  - 2.2 Task Management
  - 2.3 Notifications
-

## Module 2

service**now**

### 2.1 User Administration

- Users
- User Authentication
- Groups
- Roles
- Lightweight Directory Access Protocol (LDAP)
- Single Sign-On (SSO)



### Users

A user is one record stored in the User [sys\_user] table

Users can view their user record in **Self Service > My Profile**

Users are authenticated by various methods, including:

- Local database
- Multifactor
- LDAP
- SAML 2.0
- OAuth 2.0
- Digest Token



### Groups

A group is one record stored in the Group [sys\_user\_group] table

A group is a collection of users who share a common purpose: approve change requests, have incidents assigned to them, receive e-mail notifications, etc.

Examples: Training Group, Service Desk, Chat Support

Manage the individuals who can access ServiceNow by defining them as users in the platform.

**NOTE:** User names (represented by user IDs) are unique in ServiceNow.

The various available authentication methods include:

- **Local database:** The user name and password in their user record in the instance database
- **Multifactor:** The user name and password in the database and a passcode sent to the user's mobile device that has Google Authenticator installed
- **LDAP:** The user name and password are accessed via LDAP in the corporate directory, which has a matching user account in the ServiceNow database
- **SAML 2.0:** The user name and password configured in a SAML identity provider account, which has a matching user account in the database
- **OAuth 2.0:** The user name and password of OAuth identity provider, which has a matching user account in the database
- **Digest Token:** An encrypted digest of the user name and password in the user record

User credentials are matched to different saved credentials for each method. Multiple Provider SSO allows you to choose/use several identity providers (IdPs) to manage authentication as well as retain local database authentication.

A group is part of the user hierarchy, and a user is part of a group. Groups may be imported from a corporate directory (LDAP) or created manually in ServiceNow.

## What is a Role?

servicenow

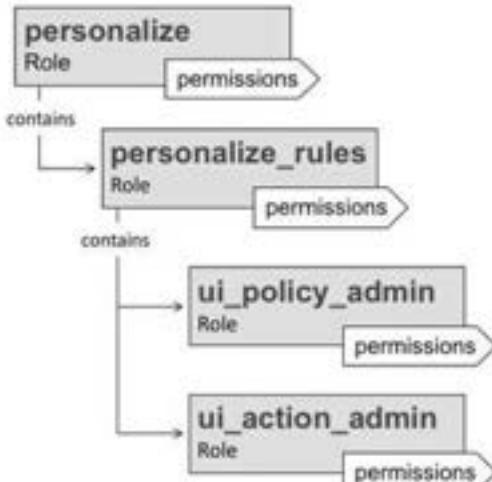
A role is one record in the **Role [sys\_user\_role]** table

A role is a group of permissions

Each role grants permissions to perform actions in parts of the platform

A role is a persona that is assigned to a group or a single user; a user can be assigned more than one role

A role may contain other roles



A role is a collection of permissions used to:

- Grant access to applications and other parts of the platform
- Assign security rights

Once access has been granted to a role, all of the groups or users assigned to that role are granted the same access.

Additionally, a role may contain other roles and any access that is granted to one role is automatically granted to any role that contains it.

In this example, the **personalize** role is able to personalize forms, lists, rules, controls, and scripts. It has its own permissions and also contains the **personalize\_rules** role.

The **personalize\_rules** role has its permissions and contains both the **ui\_policy\_admin** and **ui\_action\_admin** roles.

The **ui\_policy\_admin** role can manage UI Policies. The **ui\_action\_admin** role can manage UI Actions.

Taken with all of these relationships, the **personalize** role contains all of the roles below it in the hierarchy. However the **ui\_policy\_admin** and **ui\_action\_admin** roles do not contain the permissions of the roles above them in the illustration.

## Provided Roles

service<sup>now</sup>

 System Administrator	 Specialized Administrator	 Fulfiller	 Approver	 Requester
The <b>admin</b> role provides access to all platform features, applications, functions, and data	Specialized Administrator roles manage specific functions or applications, such as Assignment Rules, Knowledge Base, Reports, or Web Services	Users with the <b>itil</b> role may fulfill ITIL activities associated with the ITIL workflow, including Incident and Change Management	Users with the <b>approver_user</b> role can perform all requester actions and may view or modify approval records directed to the approver	Also known as <b>Employee Self Service (ESS)</b> users, these users have no roles but can submit and manage their own requests, access public pages, etc.

The System Administrator (**admin**) role has almost all roles and access to all platform features, functions, and data, with some exceptions such as HR and Security Operations constraints. **Grant this privilege carefully.**

Users holding the **admin** role can create and modify user roles, as well as impersonate other users. However, not even users with the **admin** role can impersonate a **security\_admin** role user and elevate privileges while impersonating to access higher security functionality.

The **impersonator** role can be assigned to a user to allow impersonation of other users, excluding admins, for testing and visibility purposes.

**Specialized Administrator** roles have broad access but generally manage specific functions or applications.

**Fulfiller/Process** users have clearly defined paths and workflows in the platform and have one or more roles, including the **itil** and **approver\_user** roles. They can access all functionality based on assigned roles.

**Approvers** have the **approver\_user** role, but no other roles.

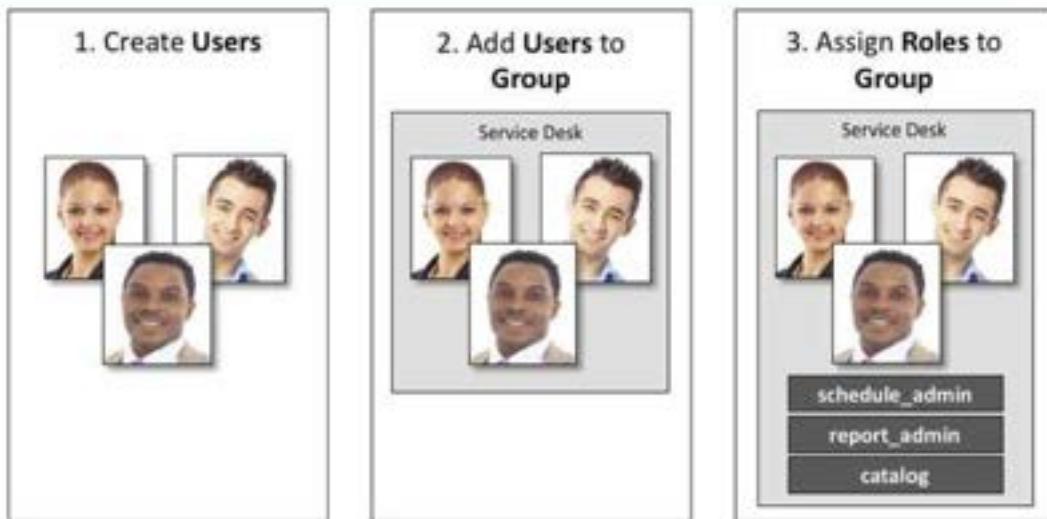
**Requesters** use the Service Catalog and Self Service applications. They can make requests only on their own behalf, and are not assigned roles.

## Users and Tasks

© 2017 ServiceNow, Inc. All Rights Reserved

## Organizing Users, Groups, and Roles

service<sup>now</sup>



Users can belong to more than one group. Groups identify a subset of users based on roles, assign permissions to approve, change, or resolve incidents and requests, provide a reference for alerts and notifications, and receive email notifications.

Every user belonging to a group inherits that group's roles, so the preferred method of role management is:

- assign users to group
- apply roles onto groups

When removing a user from a group, roles inherited by that group are revoked for that user.

Similarly, a group may contain other groups, where a child group inherits all roles owned by its parent. Users added to child groups gain roles of that child group plus any parent groups.

**Groups** include the following characteristics:

- Users can belong to more than one group
- Groups make permissions management easier
- Groups are assigned roles

**NOTE:** Group names are unique in ServiceNow.

## Lightweight Directory Access Protocol (LDAP)

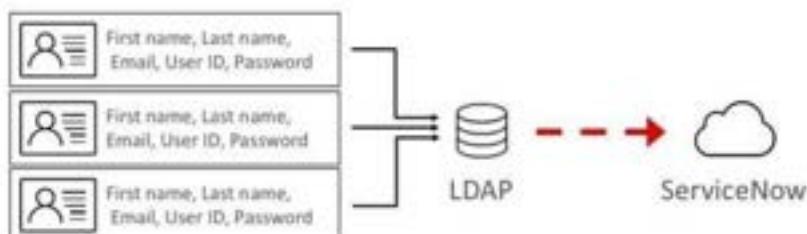
servicenow

**Lightweight Directory Access Protocol (LDAP)** can be used to streamline the user login process and automate administrative tasks such as user creation and role assignment

ServiceNow integrates with a customer's directory via LDAP

Two separate aspects of the integration include:

- Data Population
- Authentication



### LDAP Integration Aspects:

- **Data Population:** ServiceNow LDAP integration allows one to quickly and easily populate ServiceNow with user records from an existing directory. In case of data inconsistencies, configuration settings provide the ability to create, ignore, or skip records. User data can be refreshed in a timely manner via several methods including scheduled imports and the LDAP listener (only available to Active Directory deployments).
- **Authentication:** By authenticating against a customer directory, users can utilize the same credentials for the ServiceNow platform that they use for other internal resources on a company domain. Also, existing password and security policies that are already in place can be leveraged (for example: account lockout after a number of failed logins and password expiration dates). Since the ServiceNow application is receiving a "yes" or "no" from the directory, where these policies are enforced.

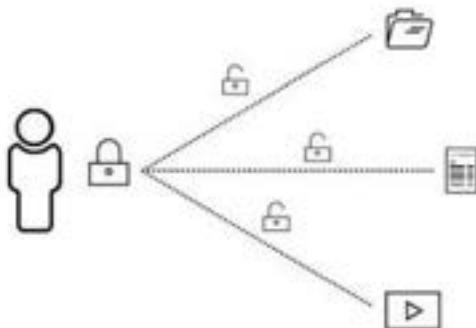
**NOTE:** The service account used to access the directory only needs read access, since ServiceNow does not modify customer data via this integration.

## Single Sign-On (SSO)

servicenow

**Single Sign-On (SSO)**, also referred to as External Authentication, is a method of access control that enables a user to log in once and gain access to the resources of multiple software systems without being prompted to log in again.

Before you enable External Authentication in your instance, you need to decide how you want to pass authentication information to ServiceNow.



Single Sign-On (SSO) lets end users login to their company portal page and be directed to their ServiceNow instance pre-authenticated, without having to enter a username and password again. This is very beneficial in large corporate environments where managing multiple user databases is prohibitive.

The Multiple Provider Single Sign-On (multi-SSO) feature in ServiceNow, enabled by a plugin, allows organizations to use one or more SSO identify providers (idPs) to manage authentication as well as retain local database (basic) authentication.

The integration supports any combination of local and external authentication methods on a single instance:

- SAML 2.0
- Digest authentication
- LDAP
- Local database authentication

## Section Summary

- Users
- User Administration
- Groups
- Roles
- LDAP
- SSO



10 – 15 minutes

### Lab 2.1 – User Administration:

- Create an Infinity Support group under the Service Desk group
- Update the Default Infinity Employee Incident template
- Create an Infinity Security group with new users

# Lab Topics

- Create an Infinity Support group under the Service Desk group
- Update the Default Infinity Employee Incident template
- Create an Infinity Security group with new users

To prepare for the possibility of supporting multiple products at Cloud Dimensions, Winnie Reich wants a new group to be created under Service Desk dedicated to Infinity support.

## Lab 2.1 User Administration

This group will consist of existing Service Desk team members, as well as new employees – all being granted the appropriate permissions for managing Infinity incident records.

Additionally, the importance of security with Infinity is driving a decision to create an Infinity Security group, which will be led by Buster Wubbel. This group must be created and populated with members.

### Create the Infinity Support Group

1. Impersonate System Administrator.

**NOTE:** Winnie does not have the appropriate user permissions to complete the task of creating a group within a group.

2. User Administration > Groups.
3. Open the Service Desk record.
4. Next, click on the Groups tab:



5. Click New to add a new group to Service Desk.

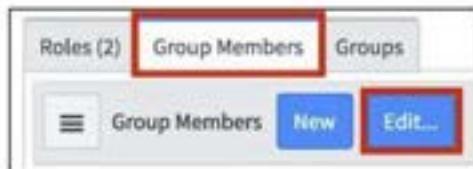
6. Fill out the record as follows:

Name: **Infinity Support**  
Manager: **Winnie Reich**  
Email: **cdisupport@cloudd.com**

7. Open the **Form Control Menu**, then select **Save** to remain in the group record.

**NOTE:** You should see a number of blue messages appear on the form confirming role provisioning.

8. Select the **Group Members** tab, then click **Edit...**



9. Add the following users to the Infinity Support group:

**Beth Anglin**  
**Kevin Edd**  
**Megan Burke**  
**Ted Keppel**

10. Click **Save**.

11. Next, under **Group Members**, create the following user by first clicking **New**, then filling out the form as follows:

User ID: **kara.prince**  
First name: **Kara**  
Last name: **Prince**  
Email: **kara.prince@cloudd.com**

**NOTE:** Creating a user from the group record will automatically add them to the group.

12. Submit.

13. The list of the Infinity Support group members should look like this:

A screenshot showing a list of users under the heading 'Group = Infinity Support'. The list includes: Ted Koppel, Megan Burke, Beth Anglin, Kevin Edd, and Kara Prince. Each entry has a magnifying glass icon and a blue circular icon with an 'i'.

### Update the Default Infinity Employee Incident Template

Due to the creation of the Infinity Support Service Desk group, per Winnie's request, the current procedure for creating new Infinity employee incident records must be modified. Specifically, the group that default Infinity employee incidents are assigned to has changed.

The Default Infinity Employee Incident template must be updated, which will be completed as the System Administrator.

1. **Incident > Open.**
2. Open any incident record and toggle on the template bar if necessary:



3. From the template toolbar, click **All templates**:

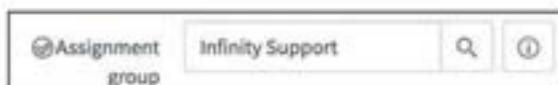


4. Next, from the menu, select **Edit** next to **Default Infinity Employee Incident** to open the template properties:

The screenshot shows a list of templates under the heading 'All templates'. There is a search bar labeled 'Filter Templates'. Below it is a table with three rows. The first row contains 'Default Infinity Employee Incident' and an 'Edit' button, which is highlighted with a red box. The second row contains 'Incident Call Type' and an 'Edit' button. The third row contains 'Major Incident' and an 'Edit' button.

All templates	
<input type="text"/> Filter Templates	
Default Infinity Employee Incident	<b>Edit</b>
Incident Call Type	Edit
Major Incident	Edit

5. Change the value for the **Assignment group** condition from Service Desk to **Infinity Support**.
6. Click **Update**.
7. Now, test and verify the changes by navigating to **Incident > Create New**.
8. From the template bar, select **Default Infinity Employee Incident**.
9. Confirm the Assignment group is **Infinity Support**:



**NOTE:** We will not save this incident record.

## Create the Security Group

As the Service Desk group continues to refine their team and processes, a request to create a **Security group** is shared.

The Infinity Support and Security groups have agreed upon a plan to establish email notifications so that crucial security issues are not missed.

The System Administrator will first create the Security group, and then an email notification, per the requirements provided.

1. **User Administration > Users.**

There are a number of new employees that do not have user accounts in ServiceNow. We will create them while practicing the saving method of **Insert**.

2. Click **New** from the Users header:



3. Configure the User form to add the Manager field.

4. Fill out the record as follows:

User ID: **hal.stone**

First name: **Hal**

Last name: **Stone**

Email: **hal.stone@cloudd.com**

Manager: **Buster Wubbel**

5. Open the **Form Control Menu**, then select **Save** to remain on the form.

6. Modify the record's fields as follows:

User ID: **arthur.west**

First name: **Arthur**

Last name: **West**

Email: **arthur.west@cloudd.com**

7. Open the **Form Control Menu**, but this time select **Insert**:



**NOTE:** If you are not taken to the list of users, **User Administration > Users**.

8. From the **Users** list, click on the **Created** column header once to sort the records by most recently created users:



9. You should see the two users just created:

User ID	Name	Email
arthur.west	Arthur West	arthur.west@cloudd.com
hal.stone	Hal Stone	hal.stone@cloudd.com

10. User Administration > Groups.

11. Click New and fill out the form as follows:

Name: **Infinity Security**  
Manager: **Buster Wubbel**  
Group email: **cdisecurity@cloudd.com**

12. Save.

13. Once the record reloads and the related lists appear, select **Group Members**, then click the **Edit...** button to add users.

14. Add the following users to the **Security** group:

Alene Rabeck  
Arthur West  
Hal Stone  
Hans Carlan

Your **Group Members List** should look like this:

Group Members List
Security
Alene Rabeck Arthur West Hal Stone Hans Carlan

15. Save.

## LAB VERIFICATION

### Infinity Support Group

Group = Infinity Support	
	User
1	Ted Keppel
1	Megan Burke
1	Beth Anglin
1	Kevin Edd
1	Kara Prince

### Infinity Security Group

Group = Infinity Security	
	User
1	Hans Cartan
1	Hal Stone
1	Arthur West
1	Alene Babcock

*Congratulations, you have completed the User Administration lab!*

## Module 2

service*now*

### 2.2 Task Management

- What is a ServiceNow Task?
- My Work / My Groups Work
- Assignment Rules
- Presence
- Notes Tab
  - Work Notes
  - Comments
  - Activity
- Chat
- Visual Task Boards

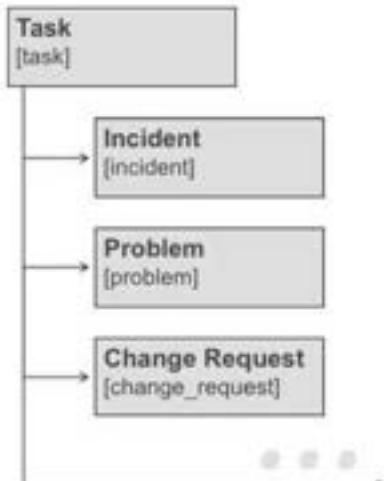
## What is a ServiceNow Task?

service*now*

A task is any record that can be assigned to a user in ServiceNow

A task is created, work is performed upon it, and eventually it moves to a closed state

Within ServiceNow, all ITIL processes (incident, problem, change, etc.) are handled through tasks



Tasks are created by users who are requesting the task to be performed, and are then updated as the task moves along a workflow. Tasks can be assigned to specific users or user groups.

The **Task [task]** table is one of ServiceNow's core tables and provides a series of standard fields used on each of the tables that extend it, such as the **Incident [incident]** and **Problem [problem]** tables. In addition, any table which extends Task can take advantage of task-specific functionality for driving tasks, including:

- **Approvals:** Approvals can be generated to a list of Approvers, either manually or automatically, according to Approval Rules. Approvals can be incorporated into workflows or can stand alone.
- **Assignments:** Assignment rules can automatically assign tasks to users or groups, ensuring that tasks are handled by the most appropriate team members.
- **Service Levels:** Service level agreements can track the amount of time that a task has been open, to ensure that tasks are completed within an allotted time.
- **Inactivity Monitors:** Inactivity monitors ensure that tasks do not fall by the wayside by notifying users when tasks have been untouched for a predefined period of time.
- **Workflow:** An administrator can specify a specific workflow process to apply to tasks that meet certain conditions. After a task is created that meets the conditions, the workflow applies an automated process to the task. The process is defined in the graphical workflow editor.

## My Work / My Groups Work

service*now*

The Service Desk application menu allows you to locate all work assigned to your group or to you specifically



My Work

List of all active tasks assigned to you

My Groups Work

List of all active tasks assigned to your group but  
**not yet** to an individual

You can quickly locate all work assigned to your group or to you specifically in ServiceNow, using the Service Desk application.

When an active task (which might include work such as incidents, problems, changes, and more) is routed to your group, it can be located under the **Service Desk > My Groups Work Module**. From there, a group member or manager may assign the task to an individual within the group.

At that point, locate any active task specifically assigned to you under the **Service Desk > My Work module**.

## Assignment Rules

servicenow

**Assignment rules** automatically set a value in the `assigned_to` and `assignment_group` fields

Conditions defined in the assignment rule determine when the rule will trigger and what values it will set

**Assignment data lookup rules** may also be created to set a value in the `assigned_to` and `assignment_group` fields, but apply only to incident records

The top screenshot shows the 'Assignment Rule' configuration page. It has tabs for 'Assign To', 'Script', and 'Conditions'. Under 'Table', 'Incident (Incident)' is selected. Under 'Conditions', there are buttons for 'Add Filter Condition' and 'Add 'OR' Clause'. A 'choose field' dropdown is shown. The bottom screenshot shows the 'Assignment Data Lookup' configuration page. It has tabs for 'Assign To', 'Script', and 'Assignment Data Lookup'. It includes fields for 'User' and 'Group'. Below these are several dropdowns and input fields for 'Category', 'Subcategory', 'Configuration Item', 'Location', 'Assignment Group', and 'Assigned To'.

Assignment rules allow you to automatically set a value in the `assigned_to` and `assignment_group` fields when a set of conditions occurs.

Create one by navigating to the **System Policy > Rules > Assignment** module.

An assignment rule must also meet these additional criteria to run:

- The task record has been created or updated. Assignment rules do not apply to unsaved changes on a form.
- The task record must be unassigned. The record cannot have an existing value for either the `assigned_to` or `assignment_group` fields. Assignment rules cannot overwrite existing assignments (including assignments set by a default value or a previously run assignment rule).
- The assignment rule is the first rule that matches the table and conditions. If more than one assignment rule matches the conditions, only the rule with the lowest order value runs

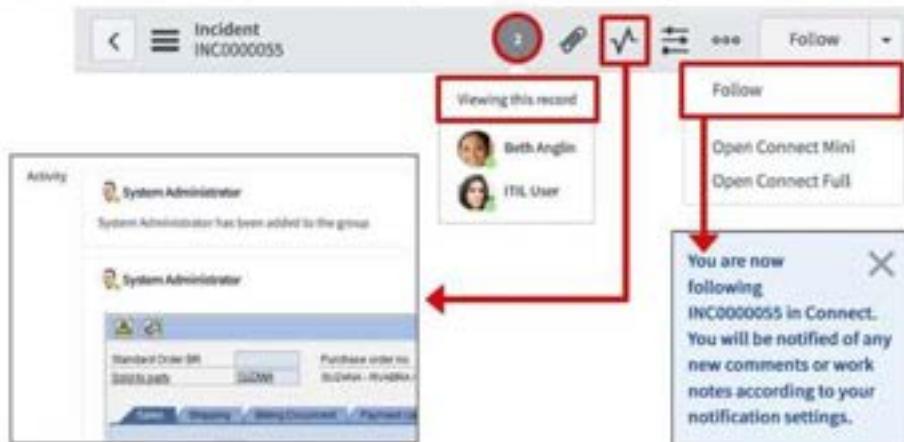
Additionally, assignment rules can be scripted, giving even more flexibility on the trigger and outcome.

Assignment data lookup rules can be created by navigating to the **System Policy > Rules > Assignment Lookup Rules** module. These rules only apply to incident records and have less options, compared to the other assignment rules.

## Presence

servicenow

The **Presence** feature facilitates synchronous collaboration within one record. You can see who is online, view their current status, and what they are viewing or editing, all in real-time.



Imagine a scenario in which you have a critical issue, documented in a Priority 1 record. Multiple stakeholders may need to view and update the record simultaneously. The **Presence** feature facilitates that collaboration, showing you who is viewing the record, displaying the record activity stream, and even allowing you to customize notifications alerting you to record updates.

The number of active viewers is listed in the form title bar. Click for a list of viewers.

**NOTE:** If you do not see this icon, you are the only viewer on this record.

Click the **Show Activity Stream** icon to jump to the record Activity section, which includes the record history and updates by you and other viewers.

Follow the record to receive notifications when the record is updated. Open **Connect** to customize these notification or start a real-time conversation with record viewers and other stakeholders.

## Presence: Real-Time Editing

servicenow

Edit records in real-time, and see edits saved by other users (just like Google Docs) which improves collaborative efforts

Real-time editing is an extension of Presence. It allows you to work with others on the same record, indicating their state (editing or viewing) as well as what their edits are (shown through the blue "pulse" icons in this screen capture).

Connect enables you to work seamlessly with others using entirely different interfaces or devices to perform tasks. You can use Connect on Visual Task Boards (VTBs), Connect Chat, Apple Watch, and more.

Connect is about working with others in real-time; reducing record resolution from days to mere minutes.

## Presence: Activity Stream Inline Editing

servicenow

Contribute to actual work inside activity streams, without drilling into the records themselves, getting work done more quickly



Just like real-time editing on a form, inline commenting on the activity stream means you can annotate active records as updates are made, allowing multiplied efforts across several pieces of work simultaneously.

To do so, navigate to a list of active task records, then:

1. Click **Show activity stream** in a flyout window from the list header
2. With the window open, scroll down to browse the records recently updated and hover over an update you wish to comment, then click **Comment**
3. Enter your comment into the text field, then click the **Post** button

A benefit of activity stream inline editing is that you are able to update multiple active records without having to open a single record.

## Notes Tab: Work Notes and Additional Comments

servicenow

Use the **Notes** tab to document task activities throughout its lifecycle and communicate with the record's requester

The **Show all journal fields** icon (≡) allows you to display multiple fields under the Notes tab, including **Work notes** and **Additional comments**

Once enabled, you can then click the **Show one journal field** icon (≡) to only display the **Work notes** field

The **Notes** tab allows you to document task activities throughout its lifecycle for both an external and internal audience. Depending on the task record type, additional fields may be available to accomplish similar outcomes including, but not limited to, the **Additional comments** field. In the example seen here, an incident record's **Notes** tab is displayed.

1. The **Work notes** field provides a log to document all the technical and behind-the-scenes work on a task. Upon saving, Work notes are stored in the record Activity section, where they can be viewed and added upon by users with permissions to view the record. Fully documenting Work notes is beneficial for Knowledge Management and critical for continuity in the task management process. Work notes are only visible to Fulfillers and are not available to external users or customers.
2. Use the **Additional comments (Customer visible)** field to communicate back and forth with the requester and other stakeholders directly in ServiceNow. For example, you may want to keep the customer apprised of progress on their record or request additional information. Upon saving, the Additional comments (including the updated information and comments history) are emailed directly to the requester. When the requester receives an email notification containing Additional comments, they can respond directly to the email and their feedback will be documented in the **Activity** log of the record, along with your Additional comments.

**NOTE:** When responding to an email from ServiceNow, do not change the Subject as it may not be saved to the correct record.

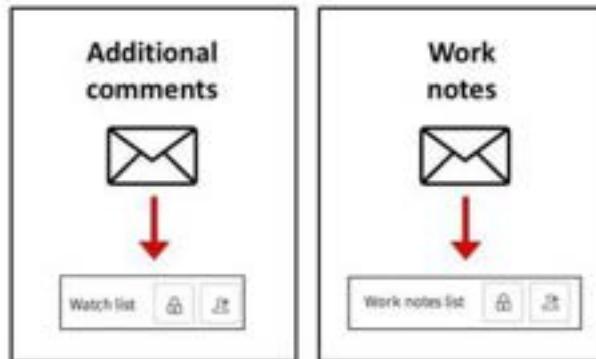
## Notes Tab: Add Stakeholders

service<sup>now</sup>

The **Watch list** and **Work notes list** enable you to send additional notifications to other users

Users added to the **Watch list** will receive anything entered as **Additional comments (Customer visible)**

Users added to the **Work notes list** will receive anything entered as **Work notes**



**NOTE:** One or both of these fields will appear on the record, under the Notes tab, depending on the record type. Incident records contain both the **Watch list** and **Work notes list** for use.

## Notes Tab: Activity

servicenow

Also located under the Notes tab is the **Activity** section

Activity provides a complete history of the record

Clicking the filter icon (▽) allows the activity information to be filtered

The screenshot shows the 'Activity' section of a ServiceNow incident record. It displays a list of activity entries, each with a user icon, a timestamp, and a brief description. The first entry is highlighted with a yellow background. To the right of the list is a vertical filter menu with a funnel icon at the top. The menu lists various categories with checkboxes, and most are checked. The categories include: All, Additional comments, Assigned to, Attachments, Configuration item, Impact, Incident state, Opened by, Priority, Relationship changes, Resolution code, and Resolution notes.

When?	Who?	What?
2017-05-04 13:48:06	ITIL User	The SAP Human Resources application is not accessible.
2017-05-04 13:48:06	ITIL User	Current access to this application has been fully restored.
2017-05-04 13:48:06	ITIL User	Assigned to: Both angles Configuration item: SAP Human Resources Impact: 2 - High Incident state: In Progress Opened by: ITIL User Priority: 1 - Critical
2017-05-04 13:48:06	ITIL User	

From creation through to closure, the entire history of the Incident record is automatically tracked and recorded in the Incident **Activity** section, located in the **Notes** tab.

The Activity section, which is read-only, documents when a change was made and by whom. These changes include assignment and reassignment, Additional comments and Work notes, updated field values, State changes, and more.

The funnel icon in the top-right of the Activity section allows you to filter your view to see only your desired categories of information.

## Connect Chat and Workspace

service<sup>now</sup>

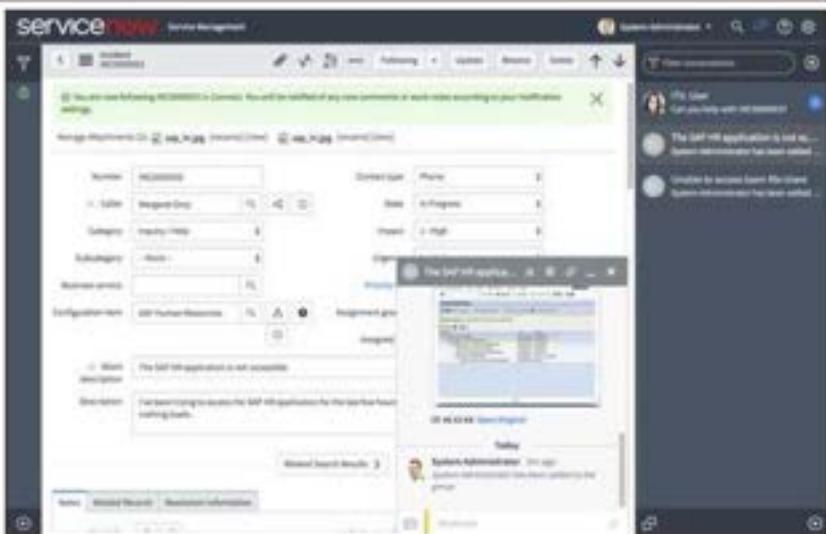
### Chat

**Connect Chat** is a real-time messaging tool that lets you work with others right where the conversation happens.

### Workspace

The **Connect Workspace** provides a full-screen view of all your Connect Chat and Connect Support conversations in one place.

Click the **Open Connect standard interface** icon (□) to open the Workspace



ServiceNow provides several tools to help you communicate with your team in real time, based on information within the platform. **Connect Chat** is accessible from the **Collaborate > Connect Chat** Module (opens full view in new tab) or from the **Connect** icon in the Banner (opens a sidebar within Content Frame). You can create new conversations with individual ServiceNow users or create custom chat groups. A green dot indicates participants who are currently online. Additional options allow you to add attachments to the chat, customize your notifications to stay in the loop on the conversation, and easily view and update related records.

The Connect Workspace provides a full-screen view of all your Connect Chat and Connect Support conversations in one place, plus additional tools to help keep track of important information in conversations. To open the Connect Workspace, navigate to **Collaborate > Connect Chat** or click the **Open Connect standalone interface** icon from the Connect Sidebar. If you do not have any recent conversations, a screen appears with helpful information about Connect.

The Connect Workspace interface is made up of three major components:

- **Connect Sidebar:** Provides access to conversations. The Connect Sidebar behaves the same way in the Workspace as it does in the Connect overlay. The only difference is that the sidebar appears on the left edge of the Connect Workspace.
- **Conversation Pane:** Displays the currently selected conversation
- **Conversation Tools:** Provide quick access to key information, conversation members, attachments, and notification preferences for the currently selected conversation. Some of the conversation tools vary depending on the type of conversation.

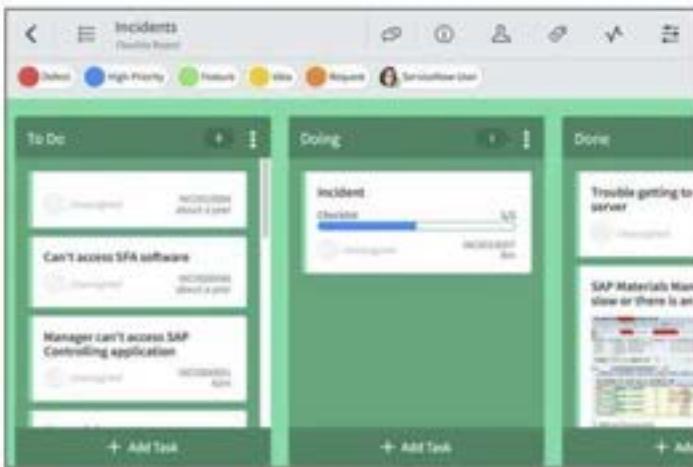
## Visual Task Boards

service<sup>now</sup>

Transform your lists and forms into an interactive graphical experience using **Visual Task Boards (VTB)**

Visual Task Boards allow you to:

- Manage your Tasks through a visual, drag-and-drop interface
- Identify process bottlenecks at a glance, in real time
- Track embedded activity screens to view updates all in one place



Use **Visual Task Boards (VTB)** to create your personal to-do list, collaborate in real-time with group members on assignments, and more. Displayed graphically as lanes and cards, VTBs provide a landing page to view and organize all your work in ServiceNow.

There are three types of VTBs:

- **Freeform:** Use Freeform boards as your personal organizer, creating individual tasks of any kind and freely adding, removing, and modifying cards and lanes
- **Guided boards:** Alternatively, use a Guided board, which is created from a list and uses a field value (e.g. Incident States) as lanes. Records in that list, which appear as cards, are actually modified when you edit cards or change lanes in a Guided board.
- **Flexible:** Flexible boards are also created from a list but lane changes do not update underlying task data

To get started with VTB, navigate to **Self-Service > Visual Task Boards** and follow the instructions for creating a board.

## Section Summary

- What is a Task?
- My Work / My Groups Work
- Assignment Rules
- Presence
- Notes Tab
- Chat
- Visual Task Boards

### Lab 2.2 Task Management



Pages 120 – 128



15 – 20 minutes

#### Lab 2.2 – Task Management:

- Create and validate assignment rules
- Browse the Service Portal Knowledge Base
- Submit an incident from the Service Portal
- Explore chat
- Fill out an incident record, using Work notes and comments

**Lab Dependency** – This lab requires the completion of Lab 1.3.

# Lab Topics

- Create and validate an assignment rule
- Browse the Service Portal Knowledge Base
- Submit an incident from the Service Portal
- Explore chat
- Fill out an incident record, using Work notes and comments

**Lab Dependency** – This lab requires the completion of Lab 1.2.

In addition to internal testing of Infinity, Cloud Dimensions has also begun to allow major partners to test devices.

An Infinity Customer Support group has been created under the Cloud Dimensions Customer Support group so that agents may assist its partner's users if they have any problems during testing.

These partner users have been instructed to browse the Cloud Dimensions Service Portal's Knowledge Base first, before submitting an incident.

The Infinity Customer Support group has defined a process but wants to ensure efficiency in incident resolution. To do this, they will create an assignment rule to properly route incoming partner Infinity incidents.

## Lab 2.2 Task Management

### Create an Assignment Rule

1. System Policy > Rules > Assignment.
2. New.
3. Fill out the form as follows:

Name: **Infinity Customer Incidents**

Under the **Applies To** tab, add the following condition:  
**Short description | contains | Infinity**

Under the **Assign To** tab,  
**Group: Infinity Customer Support**

4. Submit.

## Discussion

With a partner, discuss the rule created and what it will do when activated.

If you are attending a virtual session, share your thoughts in the chat.

## Browse the Service Portal

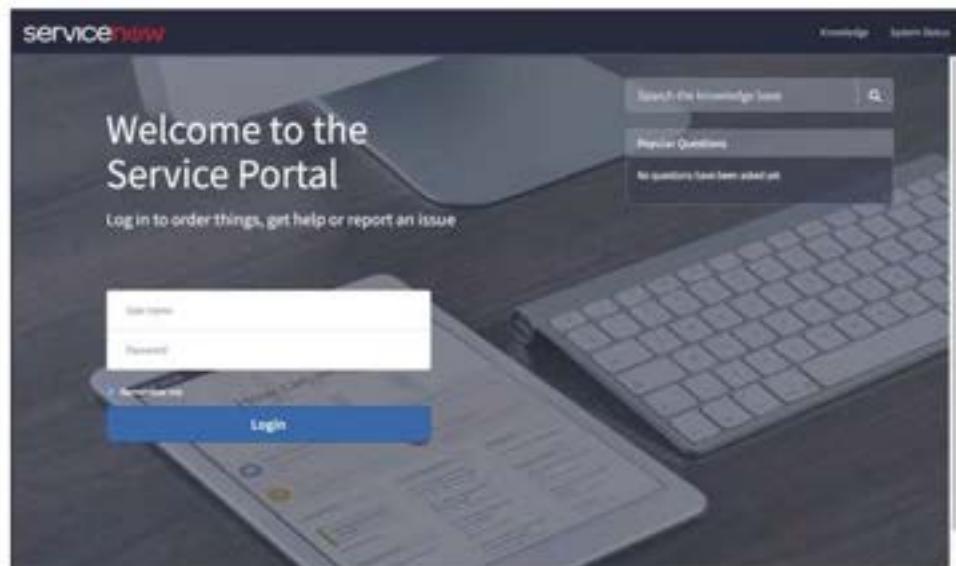
TIP: Open up another browser to complete these steps.

1. If you're not using a different browser, log out of your instance.
2. Add **/sp** to the end of the URL:

<https://instance-####.lab.service-now.com/sp>

3. Press **Enter** on your keyboard.

You should be brought to the Service Portal:



4. Log in using the following Cloud Dimensions partner user's credentials: **jon.floyd** / **floydpass**

- Type infinity not projecting into the How can we help? search bar on the home page; then press enter on your keyboard.

No Results will display:

The screenshot shows a search interface with a sidebar on the left containing links for Home, Search, All (which is highlighted in blue), Knowledge Base, Questions and Answers, and Service Catalog. The main content area displays a 'No Results' message with the text: 'Your search - infinity not projecting - did not match any documents.' Below this, there is a 'Suggestions:' section with two items: 'Make sure all words are spelled correctly' and 'Try different, more general, or fewer keywords.'

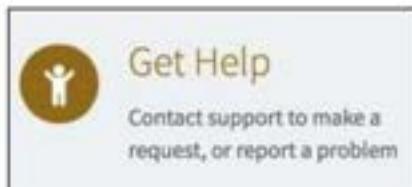
**NOTE:** Typing infinity or Infinity will result in No Results.

At this time, we will assume Jon Floyd continued browsing the Knowledge Base, but was unable to discover a solution for the issue he is experiencing.

As established by partner procedure, he will now submit an incident to receive help from Cloud Dimensions.

- Click the ServiceNow logo in the top-left corner to return to the homepage.

- Select Get Help:



- Scroll down and select Create Incident.

- Fill out the form as follows:

**Urgency: 3 – Low**

Please describe your issue below: **The Infinity is having trouble displaying clearly.**

- Click Submit (you may have to scroll to the bottom of the page).

- You will briefly see a message on the top of the screen indicating your incident was created and your profile badge should show one Request.

12. Log out of the instance as **Jon Floyd**.

**NOTE:** If you opted to use another browser for these steps, return to the original session and move to step 2 of the **Update the Incident** section below. Otherwise, continue on to the next step.

13. From your instance URL, remove the **/sp** suffix.

14. Press **Enter** on your keyboard to return to the normal login screen:

The screenshot shows the ServiceNow login page. At the top left is the Cloud Dimensions logo. Below it is a large input field for 'User name'. Underneath is a smaller input field for 'Password'. To the right of the password field is a checked 'Remember me' checkbox. At the bottom left of the form are links for 'Forgot Password?' and 'Login'. Below the login form, there is a footer section containing the text: 'About Cloud Dimensions ServiceNow', 'Welcome to the home of Infinity! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.'

## Update the Incident

1. Log into the instance as the **System Administrator**.

**NOTE:** If you experience issues with the interface caching, refresh the browser and click on the ServiceNow logo.

2. Impersonate **Rita Center**.

**NOTE:** Rita is a Customer Support Agent on the Infinity Customer Support group specializing in software-related issues.

3. **Service Desk > My Groups Work**.

4. Notice Jon's incident appears, after being assigned to the Infinity Customer Support group by the assignment rule.

**NOTE:** The assignment rule Short description condition is case-sensitive. In other words, if the short description was "The infinity ..." then the rule would not have automatically routed the incident correctly. Additional conditions, or using a script in the assignment rule could help catch these variations.

5. Open the incident record and update it as follows:

Category: Infinity  
Assigned to: Rita Center

6. Save the record, staying on the form.
7. Assume Rita has remotely accessed Jon's device and found no apparent issues.
8. Update the incident record to inform Jon it must be escalated to another team:
  - From the Notes tab, type into the **Work notes** field: **Hello Jon, I wasn't able to find anything on my end but this incident will be escalated appropriately.**
  - Check the **Additional comments (Customer visible)** checkbox
9. Click the **Post** button

The screenshot shows the 'Notes' tab of an incident record. At the top, there are tabs for 'Notes', 'Related Records', and 'Resolution Information'. Below these are two sections: 'Watch list' and 'Work notes'. The 'Work notes' section contains a yellow box with the text 'Hello Jon, I wasn't able to find anything on my end but this incident will be escalated appropriately.' To the right of this text are three icons: a magnifying glass, a clipboard, and a 'Post' button. A red circle labeled 'a' is over the magnifying glass icon. A red circle labeled 'b' is over the clipboard icon. A red circle labeled 'c' is over the 'Post' button. Below the 'Work notes' section is a checkbox labeled 'Additional comments (Customer visible)' which is checked.

9. The comment now appears under the **Activity** section:

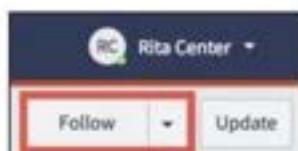
The screenshot shows the 'Activity' section of the incident record. It displays a single activity entry. The activity details are: User: 'RC - Rita Center', Date: '2017-07-19 13:00:19', and Description: 'Hello Jon, I wasn't able to find anything on my end but this incident will be escalated appropriately.'. The entire activity entry is enclosed in a red border.

10. Update the incident as follows:

Subcategory: **Hardware**  
Assigned to: **Trey Tout**

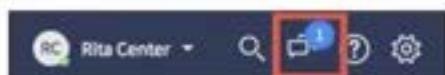
11. Save.

12. From the form header, click the **Follow** button:



Rita will now receive notices when any new comments or work notes are added. Additionally, Rita can now use chat to facilitate the resolution of this incident with peers.

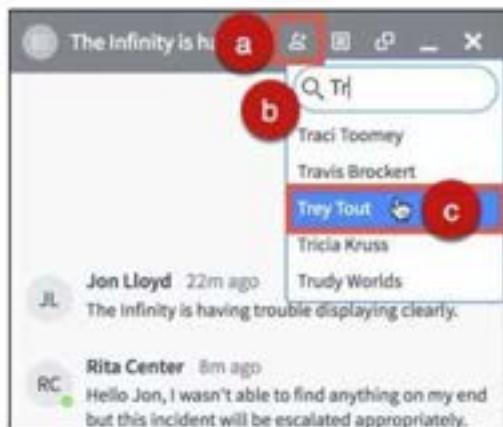
13. Open the **Connect Sidebar** to locate the conversation around the incident:



14. Select the "Infinity is having trouble" conversation from the Connect Sidebar to open a chat window.

15. Add **Trey Tout** to the conversation:

- Click the **Add User** icon
- Search for Trey Tout
- Select his name



16. With Trey added to the conversation, type the following message into the **Work notes** text field at the bottom of the chat window:

**Hello Trey, I thought you could help with this as there are no software issues detected. Thanks!**

17. Press **Enter** on your keyboard to send the message.

## Update the Incident

1. Impersonate **Trey Tout**.

**NOTE:** Trey is an Engineer on the Infinity Customer Support group. If you are unable to find his name while impersonating Rita, switch back to the System Administrator before impersonating again.

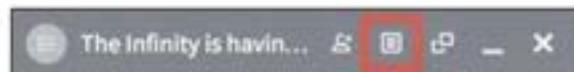
2. Open the **Connect Sidebar** to view the conversation from Rita:



3. **Service Desk > My Work**.

4. Open the incident.

**NOTE:** You could also access the incident directly from the chat window by clicking on the **View Document** icon on the chat window header:



5. If open, collapse the chat window and close the Connect Sidebar.

- With the record open, under the **Notes** tab, click the **Show all journal fields** icon to the right of **Work notes**:

The screenshot shows the 'Notes' tab of a ServiceNow incident record. The 'Work notes' section is visible, containing the text 'Work notes'. To the right of this text is a small icon of a document with a red border, which is the 'Show all journal fields' icon. Below the 'Work notes' field is a checkbox labeled 'Additional comments (Customer visible)' and a 'Post' button.

- Update the record as follows:

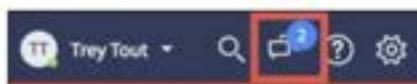
**Work notes:** Not enough information provided, contacting the customer to request more.

**Additional comments (Customer visible):** Hello Jon, I've been added to help resolve your issue. Could you please provide additional information about what you're experiencing? Thank you.

Your fields should look like this:

The screenshot shows the 'Notes' tab with updated content. The 'Work notes' field now contains the message 'Not enough information provided, contacting the customer to request more.'. The 'Additional comments (Customer visible)' field now contains the message 'Hello Jon, I've been added to help resolve your issue. Could you please provide additional information about what you're experiencing? Thank you.' The 'Show all journal fields' icon is no longer highlighted.

- Click the **Post** button to add these comments to the incident.
- Notice the Connect Sidebar icon has changed, indicating two new messages have been received:



- Set the record's **State** to **On Hold**.
- Select **Awaiting Caller** as the **On hold reason**.
- Save the record.

**NOTE:** Although the incident was not resolved at this time, Jon will be able to respond to Trey's inquiry and provide more information to identify a solution.

## Review the Record Activity

Review the updates made to the Activity section, recognizing internal (work notes) and external (additional comments) communication.

## LAB VERIFICATION

### Assignment Rules

Name	Infinity Customer Incidents	Application	Global	
Active <input checked="" type="checkbox"/>				
<b>Applies To</b> <input type="radio"/> Assign To <input type="radio"/> Script				
Table	Incident [incident]			
Conditions	<input type="button"/> Add Filter Condition <input type="button"/> Add "OR" Clause			
Short description <input type="button"/> contains <input type="text"/> Infinity <input type="button"/> AND <input type="button"/> OR <input type="button"/>				
<b>Applies To</b> <input type="radio"/> Assign To <input type="radio"/> Script				
User	<input type="text"/> <input type="button"/>			
Group	Infinity Customer Support <input type="button"/> <input type="button"/>			

### Updates to an Incident

TJ: Trey Taut	2017-07-19 13:17:09
Hello Jon, I've been added to help resolve your issue. Could you please provide additional information about what you're experiencing? Thank you.	
TJ: Trey Taut	2017-07-19 13:21:59
Not enough information provided, contacting the customer to request more.	
RK: Rita Carter	2017-07-19 13:01:18
Hello Trey, I thought you could help with this as there are no software issues detected. Thanks!	
RK: Rita Carter	2017-07-19 13:01:28
Assigned to: Trey Taut was Rita Carter	
RK: Rita Carter	2017-07-19 13:00:28
Hello Jon, I wasn't able to find anything on my end but this incident will be escalated appropriately.	

**Congratulations, you have completed the Task Management lab!**

### Users and Tasks

© 2017 ServiceNow, Inc. All Rights Reserved

## Module 2

service*now*

### 2.3 Notifications

- What is an Event?
- What is a Notification?
- Tasks for Email Setup
- Creating Notifications
- Dot-Walking
- Subscriptions

## Events and Notifications

service<sup>now</sup>

Preview Email

Incident has been closed.

Summary details

Closed by: System Administrator

Closed notes: Fixed

You can view all the details of the incident by following the link below:

[Take me to the Incident](#)

Thank you.

NHNSC00000004

### Events

An **event** is an indication to the ServiceNow processes that something has occurred

Events are triggered by:

- **User actions:** Logging in, approving a request, renaming an attachment, etc.
- **Scripts:** Business Rules and Workflows

### Notifications

**Notifications** can be triggered by events in the platform, and require no scripting knowledge

All baseline events have built in logic to respond when an event occurs. Possible responses include making a change to a record in the database, creating a new record, sending a notification, or logging a message. The event definitions are in the Event Registry [**sysevent\_register**] table. The Event Log displays records from the Event [**sysevent**] table. To see a log of every generated event navigate to **System Policy > Events > Event Log**. By convention, events are named using the syntax <table name>. <unique event name>. For example, **incident.updated**, or **problem.closed**.

**NOTE:** Use email notifications to send selected users email about specific activities in the platform, such as updates to incidents or change requests.

A Notification is a tool for alerting users that events that concern them have occurred, including the following methods:

- Email
- SMS
- Meeting Invitation

Notifications are received by configured users and voluntary recipients.

The platform enables you to send Notifications to email addresses of your choice and to any SMS capable device (such as a mobile phone).

## Notifications: Email Setup Tasks

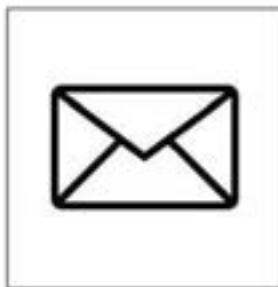
servicenow



1. Configure Email Properties



2. Create Notifications



3. Configure Email Addresses and Subscribe to Notifications

The tasks required for setting up email notifications includes:

1. Configure SMTP and POP server settings in the email properties. The default settings are intended for the ServiceNow mail server and do not have to be changed unless you intend to use your own SMTP or POP server.
2. Create the Email Notifications for your users. The ServiceNow platform has many useful Notifications in the base platform, and you can create new records or edit an existing notification if desired.
3. To subscribe or unsubscribe to Notifications and create additional email addresses, the Subscription Based Notifications plugin is installed by default. This is functionality that provides users with the flexibility of configuring the email notifications and delivery method. The **Subscribable** option allows all users to subscribe to this Notification.

**Note:** The system does not exclude recipients based on access controls. Recipients can receive email about records that they cannot normally access from the user interface. For example, requesters can receive email about incidents and catalog requests opened on their behalf even though they normally do not have access to these records. If a Notification includes record details, verify that all recipients need these details. If the record contains sensitive or protected data, consider restricting the recipient list to just those users and groups who normally have access to it, and do not enable the **Subscribable** option.

## Creating Notifications

service<sup>now</sup>

1. After choosing a name and a table, specify when to send the Notification

This screenshot shows the 'When to send' configuration screen. It has two main sections: 'Inserted' and 'Updated'. Below these sections is a 'Conditions' section with a 'Add Filter Condition' button.

2. Next, decide who will receive the Notification

This screenshot shows the 'Who to send' configuration screen. It has two main sections: 'User' and 'Group'. Below these sections is a 'Users/Groups in Fields' section with a 'Subscribe' checkbox.

3. Finally, define what the Notification will contain

This screenshot shows the 'Content' configuration screen. It features a rich text editor with various formatting tools and a preview area below it.

Use Notifications to notify users about specific activities in ServiceNow, such as updates to incidents or change requests. Notifications allow administrators to specify:

- When to send the Notification
- Who receives the Notification
- What content is in the Notification

Notifications can be sent when a record is **Inserted** or **Updated** (or both) into the **Table** specified above, only if the specified **Conditions** are met.

Notifications can be sent to specific **Users** and **Groups**. If you address the Notification to a user with an inactive record in the User [`sys_user`] table, the system does not send the notification to that user.

**TIP:** Consider limiting the recipient list of any notification to 1000 users. By default, if a Notification has more than 100 intended recipients, the system creates multiple notification messages with up to 100 recipients each. If you want to change the recipient limit, set the system property `glide.email.smtp.max_recipients`.

If using an **Email Template** then **Subject** and **Message** will be used from the template unless overridden with a **Subject** and **Message** on this form.

## Notifications: Dot-Walking

servicenow

Dot-Walking can get information from a series of referenced fields from different tables, including Users and Groups

1. From **Users/Groups in fields**, select the reference field you will dot-walk through
2. Click the **Expand Item** icon (+)
3. Select the field which holds the value to be referenced
4. Click the **Add Item** icon (>)



### Who Will Receive Notifications?

**Users and Groups** - You might send Notifications of an incident update to the person who opened the incident and the person to whom the incident was assigned by including the **Opened by** and **Assigned to** fields. You may also want to send Notifications to the group assigned to the incident by including the **Assignment group** field.

If the user or group field you want to reference is not on the current table (e.g., **Manager**), add the field by dot-walking to the table it is found on.

To achieve this, from the Notification **Who will receive** tab, click the **Unlock** icon for **Users/Groups in fields** and find the reference field under the Available column. Then click the **Expand Item** (+) icon to dot-walk through this field onto the table it is referencing. In doing so, all fields on this referenced table will be accessible to the table it is found on via a reference field.

In the example illustrated above, the **Caller** field on an incident is **David Loo**. By using dot-walking, this Notification can be sent to the caller (David Loo) of the incident, as well as the caller's manager (Bud Richman). These values are automatically populated because of fields selected – not hardcoded individual names. In other words, this Notification will automatically identify the caller and the caller's manager (if applicable) based on the information provided in the incident.

## Notifications: Subscriptions

servicenow

From **Settings**, users are able to determine which notifications they are subscribed to and will receive emails for

Users can also define additional notification channels (for example, a secondary email or push notifications through the ServiceNow mobile application)

### Short Message Service

In addition to email notifications, users can be notified by a Short Message Service (SMS), also known as a text message on mobile telephones



Subscriptions allow users to be informed of various activity occurring in the platform, whether it directly relates to them or not.

The Notifications page of the Settings is where users can define Notification channels (methods of receiving Notifications), as well as manage their subscriptions to system Notifications.

SMS (Short Messaging Service) is the standard protocol used to deliver short text messages to mobile phones.

Most mobile phones support SMS, even if they do not support more sophisticated messaging, like email.

Alerts and Notifications can be sent to SMS devices as well as standard email notifications.

Notifications to SMS devices are particularly useful when critical events require immediate attention, and waiting for an email notification to be accessed and viewed is too slow.

### Section Summary

- What is an Event?
- What is a Notification?
- Tasks for Email Setup
- Creating Notifications
- Dot-Walking
- Subscriptions



#### Lab 2.3 – Events and Notifications:

- Develop a new email notification
- Test the notification

**Lab Dependency** – This lab requires the completion of Lab 2.1.

# Lab Topics

- Develop a new email notification
- Test the notification

**Lab Dependency** – This lab requires the completion of Lab 2.1.

Buster Wubbel – manager of the Infinity Security group – has requested that a notification be created to alert his team whenever a priority 1 (P1) incident is created with a security issue being reported.

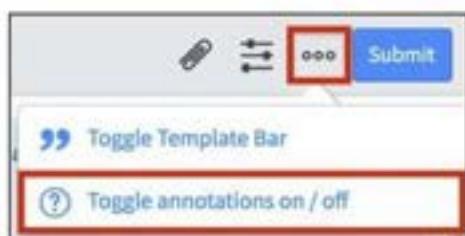
## Lab 2.3

### Notifications

Due to the urgency of resolving such an incident, the notification should also be sent to the Infinity Support group's manager in the event that the group can be used to quickly assist with a solution.

#### Develop a Notification

1. Impersonate **System Administrator**.
2. **System Notification > Email > Notifications**.
3. **New**.
4. After the new notification record loads, open the **More options** menu from the form header, then select **Toggle annotations on / off**:



**NOTE:** Annotations appear on various forms and contain useful information. It is always recommended to read them before toggling them off.

5. Set the Name to P1 Infinity Security Incident:

Name	P1 Infinity Security Incident
------	-------------------------------

The notification that will be created will inform the Infinity Support and Infinity Security groups and managers whenever a critical (Priority 1) incident is created with certain field values met.

6. Under the When to send tab, set the following values:

Inserted: [checked]

Updated: [unchecked]

Conditions:

Category   is   Infinity	AND
Subcategory   is   Security	AND
Assignment group   is   Infinity Support	AND
Priority   is   1 - Critical	

Your When to send tab should look like this:

Inserted <input checked="" type="checkbox"/>	Updated <input type="checkbox"/>						
Conditions: <a href="#">Add Filter Condition</a> <a href="#">Add "OR" Clause</a>							
All of these conditions must be met							
Category	is	\$	Infinity	AND	OR	X	
Subcategory	is	\$	Security	AND	OR	X	
Assignment group	is	\$	Infinity Support	Q,	AND	OR	X
Priority	is	\$	1 - Critical	AND	OR	X	

7. Select the Who will receive tab.

8. Click the Edit Groups icon (closed lock) on the Groups field:

Groups	
--------	--

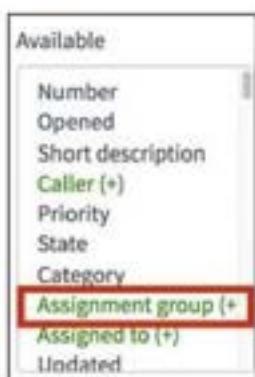
9. Search for and select Infinity Security.

10. Click the **Unlock** icon (closed lock) for **Users/Groups in fields**:



Instead of selecting a user by name, thus hardcoding that particular user to the notification, select a field that contains the data of a user account. Doing so will require dot-walking tables.

11. Highlight the **Assignment group (+)** field (formatted in green) under the **Available** bucket:



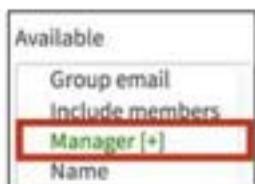
**NOTE:** The (+) icon indicates a field is a reference field, which can then be used to dot-walk from one table to another.

12. With **Assignment group (+)** highlighted, click the **Expand Item** icon (+) between the **Available** and **Selected** buckets:



**NOTE:** Doing so successfully dot-walks from the Incident table into the User table.

13. Scroll down in the **Available** list to find and then highlight **Manager [+]**:



14. Click the Add Item icon (>).

This adds **Assignment group.Manager** to the **Selected bucket** or, in other words, the notification will be sent to the manager of the incident's assignment group. Per the conditions defined, this assignment group will be **Infinity Support**.

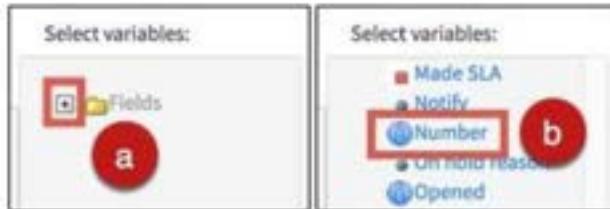
15. Click the **What it will contain** tab.

16. Type the following into the **Subject** field:

**IMPORTANT! P1 Infinity Security Incident Created**

17. Add a dynamic value placeholder to the subject:

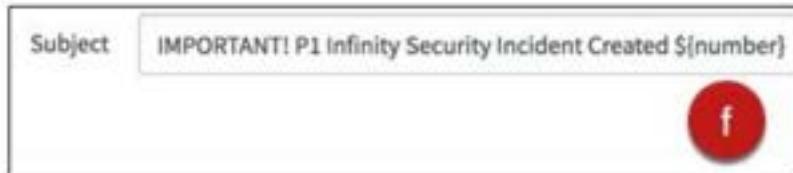
- From **Select variables**, expand the **Fields** list by clicking the + icon
- Scroll down and select the **Number** field variable



- Highlight  `${number}` from the **Message HTML** field, then copy the text to your clipboard
- Click your cursor into the **Subject** field at the end of the text
- Press the **spacebar** on your keyboard once



- f) Paste the \${number} text from your clipboard



**NOTE:** The \${number} placeholder is added to the end of the subject text and is dynamic, meaning it will automatically populate with the incident number value of the record that triggers the notification. Additional placeholders can be found and added from the **Select variables** section to the subject and Message HTML fields.

18. Update the **Message HTML** field to include the following text, replacing any existing text:

**Critical Incident \${URI\_REF} has been created with an Infinity Security category.**

**NOTE:** The \${URI\_REF} placeholder includes an upper-case "I" after the "UR" text. Check to see how this placeholder behaves after the notification is generated.

19. Click **Submit** to save the notification.

## Test and Verify the Notification

Create an incident to trigger the notification, then check to verify it was sent.

**NOTE:** Email is not enabled for the ServiceNow Lab Instance.

1. Impersonate **Kevin Edd**.
2. **Incident > Create New**.
3. Fill out the record as follows:

Caller: **Kevin Edd**  
Category: **Infinity**  
Subcategory: **Security**  
Impact: **1 - High**  
Urgency: **1 - High**  
Priority: **1 – Critical** (autofills)  
Assignment group: **Infinity Support**  
Short description: **Testing P1 Security Notification**

**4. Submit the incident.**

Because email is disabled on the instance, impersonate **System Administrator** to check the instance's email logs.

**5. Impersonate System Administrator.**

**6. System Mailboxes > Outbound > Outbox.**

**7. Locate the record with the Subject **IMPORTANT! P1 Infinity Security Incident Created INC#####**, then click on the **Created** timestamp to open the record.**

**8. Scroll down to review the **Recipients** list.**

**9. Scroll down further, then click the **Preview HTML Body** related link to display a preview of the message received by the recipients:**



**NOTE:** The \${URI\_REF} placeholder renders as a direct link to the incident record.

## LAB VERIFICATION

### Infinity Security P1 Notification

The screenshot shows the "Emails (Outbox view)" screen. The search bar indicates "Created on Today". The results table shows two entries:

	Created	Recipients	Subject
<input type="checkbox"/>	2017-09-14 23:09:14	infinitysecurity@cloudit.com, winnie.reich@cloudit.com	IMPORTANT! P1 infinity Security Incident Created INC0010012
<input type="checkbox"/>	2017-09-14 23:09:14	infinitysupport@cloudit.com	Incident INC0010012 has been opened on your behalf

**Congratulations, you have completed the Notifications lab!**

## Module Recap: Users and Tasks

servicenow

**Why** would you use these capabilities?

**When** would you use these capabilities?

**How often** would you use these capabilities?

User Administration	Task Management	Events and Notifications
User Authentication Groups and Roles LDAP SSO	My Group and My Groups Work Assignment Rules Connect Chat	Notifications Dot-Walking Subscriptions

service**now**



## Module 3

### Data Administration

- 3.1 Data Schema
  - 3.2 CMDB
  - 3.3 Import Sets
  - 3.4 Reporting
  - 3.5 Data Security
-

## Module 3

service*now*

### 3.1 Data Schema

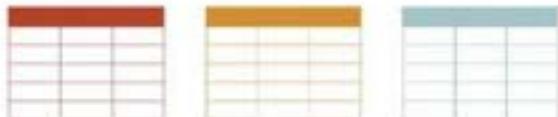
- Database Structure and Hierarchy
  - Tables
  - Records
  - Fields
- System Definition Table Modules
- Table Relationships
- Schema Map

## ServiceNow Database

servicenow



Everything in ServiceNow is built on a MySQL database, containing tables, which you access through the ServiceNow Graphical User Interface (GUI)



The **Database** contains **Tables**, **Tables** contain **Records**, **Records** contain **Fields** – a **Field** is an individual piece of data in a record

Data in ServiceNow is stored and managed according to a structure that administrators can view and configure. A table is a collection of records in the database. Each row on a table list corresponds to a record in a table, and each column on a table list corresponds to a field on that table.

Instance information is stored in database **Tables**, which contains **Records**. A **Table** is a data structure or database component and data is stored in tables. The individual pieces of data in a record are called **Fields**. When looking at data in a list view, data in these fields can be modified using the record List Editor functionality.

Lists and forms provide a friendly user interface (UI) for managing tables and records.

## Record Hierarchy

servicenow

Information is documented and saved in **fields**

Many fields make up a record, which can be viewed and edited using a **form**

One or more records make up a searchable, sortable **list**

The image contains two screenshots of the ServiceNow interface. The top screenshot shows a 'Ticket' form with various input fields. A red box highlights the 'Title' field, and a red arrow points down to the second screenshot. The second screenshot shows a 'Ticket' list view with multiple rows of data. A red box highlights the first row in the list.

Data in ServiceNow is entered into individual **fields**, which are displayed on a **form**. Once saved, these forms make up **lists**. A list typically comprises forms with a common theme, such as a record type, an assignment group, or a priority.

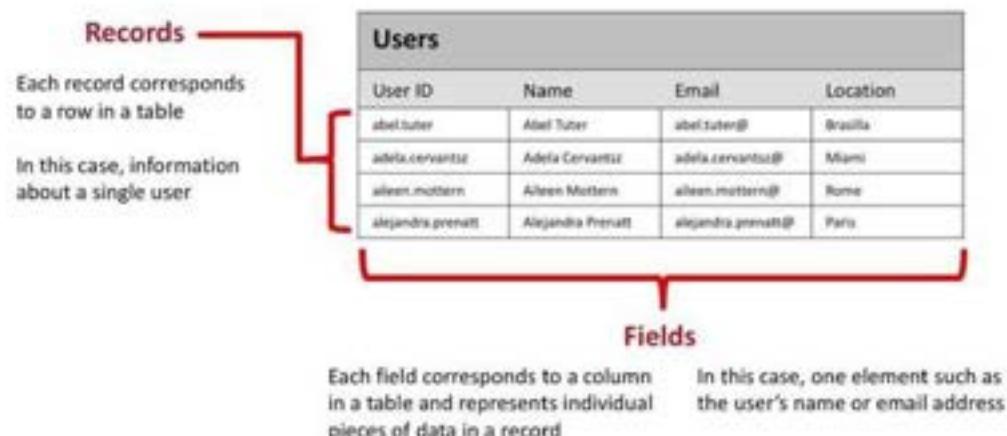
You can then sort or search in these lists to locate a particular form, which contains data in its fields.

**NOTE:** ServiceNow often uses the term **record** to describe all the data saved within a particular form. The form is simply the way that the user can view and modify that record.

## Table Components

servicenow

A table is a collection of records in the database



Records are identified by a 32-character, globally unique ID, called a `sys_id`.

Record numbers are automatically incremented, and the number format per table in the system can be changed by visiting the **Number Maintenance** application.

For example, change the default incident record number prefix from "INC" to "IN."

Fields are available in a variety of different types, including: Choice, Date/Time, Journal, Reference, and more. Field types define how a field is interacted with through the interface, as well as the type and format of data it can store.

## Field Attributes

servicenow

Each field has three attributes:  
a label, a name, and a value

Users		
User ID	Name	Location
abel.tutur	Abel Tutur	Brasilia
adela.cervantes	Adela Cervantes	Miami
aileen.mottern	Aileen Mottern	Rome
alejandra.premati	Alejandra Premati	Paris
alessandro.mascali	Alessandro Mascali	Frankfurt
alene.rabeck	Alene Rabeck	London

### Field Label

The label is a user-friendly term which allows people to identify the field in the user interface

Example: User ID, Name, Location

### Field Name

The name is a unique term that the system uses to identify the field in scripts and automated business processes

Example: user\_name, name, location

### Value

The values are actual data

Each field has three attributes: a **label**, a **name**, and a **value**.

The **field label** in this example includes User ID, Full Name, and Location

The **name** is a unique term that the system uses to identify the field in scripts and automated business processes. For example, the name attribute of the Location field on the User table is sys\_user.location. Do not confuse the name attribute with the Name field on the user table, which is a label. The name attribute of the Name field on the User table is sys\_user.name.

**NOTE:** The Name field is a combination of the First name and Last name fields of the user record.

**Values** are the actual data, such as this user's name, Aileen Mottern, or her location, Rome. In some cases, the value may be empty, or null.

## System Definition Module: Tables

servicenow

The screenshot shows the 'Tables' module in the System Definition area. The interface includes a top navigation bar with tabs for 'Tables', 'New', 'Grid', 'Label', and a search bar. Below this is a toolbar with icons for 'Add', 'Update', 'Delete', and 'Print'. A message 'All 1000 records are now visible' is displayed. The main content is a table with columns: 'Label' (with a dropdown arrow), 'Name', 'Extends table', 'Extends', and 'Updated'. The table lists various system tables such as 'JBO Application', 'JBO Load Balancer', 'Access Control', 'Access Rule', 'Addresses', 'ACE', 'AGILE', 'Action', and 'Active Cluster Transactions', each with its respective details and last update timestamp.

Label	Name	Extends table	Extends	Updated
JBO Application	cmdb_ci_appl_and_wrl	Application	None	2017-05-25 18:17:03
JBO Load Balancer	cmdb_ci_lb_wrl	Load Balancer	None	2017-05-25 18:16:07
Access Control	sys_security_ac	Application	None	2017-05-25 18:16:07
Access Rule	sys_security_ac_rule	Application	None	2017-05-25 18:16:07
Addresses	cmdb_ci_ip	Configuration	None	2017-05-25 18:17:07
ACE	cmdb_ci_lb_ip	Load Balancer	None	2017-05-25 18:16:07
AGILE	cmdb_ci_endpoint_ac	Endpoint	None	2017-05-25 18:16:07
Action	act_action	None	None	2017-05-25 18:11:09
Active Cluster Transactions	ac_transactions	None	None	2017-05-25 18:21:27

The Tables [sys\_db\_object] table contains a record for each table in the database.

To view the list of tables, navigate to **System Definition > Tables**. From there, to open the record for an existing table, click a table label.

Using the Tables module you can:

- View, add, or modify columns with a searchable and sortable embedded list, define the auto-number format, make the table extendable by other tables, and create modules for the table.
- Launch a schema map for a table by clicking the Show Schema Map related link.
- Open the dictionary entries for the table by right-clicking the form header then selecting Show Dictionary Record.
- Navigate directly to the default list or form view for the table by clicking the Show List or Show Form related link.
- Delete all records from a table by clicking the Delete All Records button on the form header.

## System Definition Module: Tables & Columns

serviceNow

Tables & Columns					
Click a button to create a new table or application, or 删除 all applications.					
<a href="#">Create Table</a>   <a href="#">Create Application</a>   <a href="#">Delete Applications</a>					
Or, select a table to browse its columns and indices.					
<b>Table Headers</b> <table border="1"> <thead> <tr> <th>Column Headers</th> <th>Column Aliases</th> </tr> </thead> <tbody> <tr> <td> <input checked="" type="checkbox"/> <a href="#">NET Application (model_id, id, app_id, net)</a>  <input checked="" type="checkbox"/> <a href="#">NET User Behavior (model_id, id, net)</a>  <input checked="" type="checkbox"/> <a href="#">Access Control (sys, resources, net)</a>  <input checked="" type="checkbox"/> <a href="#">Access Policy (sys, security, net, role)</a>  <b>REMOVED BY SPONSOR (LAW)</b>  <input checked="" type="checkbox"/> <a href="#">NET Service (id, id, net)</a>  <input checked="" type="checkbox"/> <a href="#">NET API (model_id, id, endpoint_id)</a>  <input checked="" type="checkbox"/> <a href="#">Active User (active)</a>  <input checked="" type="checkbox"/> <a href="#">Active Cluster Transactions (id, cluster, transaction)</a>  <input checked="" type="checkbox"/> <a href="#">Active Directory Domain Controller (model_id, id, controller)</a>  <input checked="" type="checkbox"/> <a href="#">Active Directory Domain Controller (domain_controllers) (endpoint_id, endpoint_id, ad_domain)</a>  <input checked="" type="checkbox"/> <a href="#">Active Directory Forest (domain_id, endpoint_id, endpoint_id, ad_forest)</a>  <input checked="" type="checkbox"/> <a href="#">Active Directory Service (model_id, id, app_id, active_directory)</a>  <input checked="" type="checkbox"/> <a href="#">Active PI jobs (pi, active, pi_id)</a>  <input checked="" type="checkbox"/> <a href="#">Active Transactions (id, transaction)</a>  <input checked="" type="checkbox"/> <a href="#">Authenticode Business Model (model_id, app_id, model, model)</a>  <input checked="" type="checkbox"/> <a href="#">Authenticode Business Model Process (model_id, id, app_id, model, model, proc)</a>  <input checked="" type="checkbox"/> <a href="#">Activity Designer (act, element, activity)</a>  <input checked="" type="checkbox"/> <a href="#">Activity Variables (act, activity, variable)</a>  <input checked="" type="checkbox"/> <a href="#">All Owners (model_id, id, app_id, domain)</a>  <input checked="" type="checkbox"/> <a href="#">All Parent (model_id, id, directory, ad_forest)</a>  <input checked="" type="checkbox"/> <a href="#">All Service Role (model_id, endpoint_id, service_id)</a>  <input checked="" type="checkbox"/> <a href="#">All Field (id, app_field_id, creator)</a> </td><td> <input checked="" type="checkbox"/> <a href="#">model_id, id, net</a>  <input checked="" type="checkbox"/> <a href="#">sys, resources, net</a>  <input checked="" type="checkbox"/> <a href="#">id, net</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, endpoint_id</a>  <input checked="" type="checkbox"/> <a href="#">id, net</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, controller</a>  <input checked="" type="checkbox"/> <a href="#">id, controller, ad_domain</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, endpoint_id, ad_forest</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, active_directory</a>  <input checked="" type="checkbox"/> <a href="#">pi, active, pi_id</a>  <input checked="" type="checkbox"/> <a href="#">id, transaction</a>  <input checked="" type="checkbox"/> <a href="#">model_id, app_id, model, model</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, model, model, proc</a>  <input checked="" type="checkbox"/> <a href="#">act, element, activity</a>  <input checked="" type="checkbox"/> <a href="#">act, activity, variable</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, domain</a>  <input checked="" type="checkbox"/> <a href="#">model_id, id, directory, ad_forest</a>  <input checked="" type="checkbox"/> <a href="#">model_id, endpoint_id, service_id</a>  <input checked="" type="checkbox"/> <a href="#">id, app_field_id, creator</a> </td></tr> </tbody> </table>		Column Headers	Column Aliases	<input checked="" type="checkbox"/> <a href="#">NET Application (model_id, id, app_id, net)</a> <input checked="" type="checkbox"/> <a href="#">NET User Behavior (model_id, id, net)</a> <input checked="" type="checkbox"/> <a href="#">Access Control (sys, resources, net)</a> <input checked="" type="checkbox"/> <a href="#">Access Policy (sys, security, net, role)</a> <b>REMOVED BY SPONSOR (LAW)</b> <input checked="" type="checkbox"/> <a href="#">NET Service (id, id, net)</a> <input checked="" type="checkbox"/> <a href="#">NET API (model_id, id, endpoint_id)</a> <input checked="" type="checkbox"/> <a href="#">Active User (active)</a> <input checked="" type="checkbox"/> <a href="#">Active Cluster Transactions (id, cluster, transaction)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Domain Controller (model_id, id, controller)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Domain Controller (domain_controllers) (endpoint_id, endpoint_id, ad_domain)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Forest (domain_id, endpoint_id, endpoint_id, ad_forest)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Service (model_id, id, app_id, active_directory)</a> <input checked="" type="checkbox"/> <a href="#">Active PI jobs (pi, active, pi_id)</a> <input checked="" type="checkbox"/> <a href="#">Active Transactions (id, transaction)</a> <input checked="" type="checkbox"/> <a href="#">Authenticode Business Model (model_id, app_id, model, model)</a> <input checked="" type="checkbox"/> <a href="#">Authenticode Business Model Process (model_id, id, app_id, model, model, proc)</a> <input checked="" type="checkbox"/> <a href="#">Activity Designer (act, element, activity)</a> <input checked="" type="checkbox"/> <a href="#">Activity Variables (act, activity, variable)</a> <input checked="" type="checkbox"/> <a href="#">All Owners (model_id, id, app_id, domain)</a> <input checked="" type="checkbox"/> <a href="#">All Parent (model_id, id, directory, ad_forest)</a> <input checked="" type="checkbox"/> <a href="#">All Service Role (model_id, endpoint_id, service_id)</a> <input checked="" type="checkbox"/> <a href="#">All Field (id, app_field_id, creator)</a>	<input checked="" type="checkbox"/> <a href="#">model_id, id, net</a> <input checked="" type="checkbox"/> <a href="#">sys, resources, net</a> <input checked="" type="checkbox"/> <a href="#">id, net</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, endpoint_id</a> <input checked="" type="checkbox"/> <a href="#">id, net</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, controller</a> <input checked="" type="checkbox"/> <a href="#">id, controller, ad_domain</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, endpoint_id, ad_forest</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, active_directory</a> <input checked="" type="checkbox"/> <a href="#">pi, active, pi_id</a> <input checked="" type="checkbox"/> <a href="#">id, transaction</a> <input checked="" type="checkbox"/> <a href="#">model_id, app_id, model, model</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, model, model, proc</a> <input checked="" type="checkbox"/> <a href="#">act, element, activity</a> <input checked="" type="checkbox"/> <a href="#">act, activity, variable</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, domain</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, directory, ad_forest</a> <input checked="" type="checkbox"/> <a href="#">model_id, endpoint_id, service_id</a> <input checked="" type="checkbox"/> <a href="#">id, app_field_id, creator</a>
Column Headers	Column Aliases				
<input checked="" type="checkbox"/> <a href="#">NET Application (model_id, id, app_id, net)</a> <input checked="" type="checkbox"/> <a href="#">NET User Behavior (model_id, id, net)</a> <input checked="" type="checkbox"/> <a href="#">Access Control (sys, resources, net)</a> <input checked="" type="checkbox"/> <a href="#">Access Policy (sys, security, net, role)</a> <b>REMOVED BY SPONSOR (LAW)</b> <input checked="" type="checkbox"/> <a href="#">NET Service (id, id, net)</a> <input checked="" type="checkbox"/> <a href="#">NET API (model_id, id, endpoint_id)</a> <input checked="" type="checkbox"/> <a href="#">Active User (active)</a> <input checked="" type="checkbox"/> <a href="#">Active Cluster Transactions (id, cluster, transaction)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Domain Controller (model_id, id, controller)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Domain Controller (domain_controllers) (endpoint_id, endpoint_id, ad_domain)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Forest (domain_id, endpoint_id, endpoint_id, ad_forest)</a> <input checked="" type="checkbox"/> <a href="#">Active Directory Service (model_id, id, app_id, active_directory)</a> <input checked="" type="checkbox"/> <a href="#">Active PI jobs (pi, active, pi_id)</a> <input checked="" type="checkbox"/> <a href="#">Active Transactions (id, transaction)</a> <input checked="" type="checkbox"/> <a href="#">Authenticode Business Model (model_id, app_id, model, model)</a> <input checked="" type="checkbox"/> <a href="#">Authenticode Business Model Process (model_id, id, app_id, model, model, proc)</a> <input checked="" type="checkbox"/> <a href="#">Activity Designer (act, element, activity)</a> <input checked="" type="checkbox"/> <a href="#">Activity Variables (act, activity, variable)</a> <input checked="" type="checkbox"/> <a href="#">All Owners (model_id, id, app_id, domain)</a> <input checked="" type="checkbox"/> <a href="#">All Parent (model_id, id, directory, ad_forest)</a> <input checked="" type="checkbox"/> <a href="#">All Service Role (model_id, endpoint_id, service_id)</a> <input checked="" type="checkbox"/> <a href="#">All Field (id, app_field_id, creator)</a>	<input checked="" type="checkbox"/> <a href="#">model_id, id, net</a> <input checked="" type="checkbox"/> <a href="#">sys, resources, net</a> <input checked="" type="checkbox"/> <a href="#">id, net</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, endpoint_id</a> <input checked="" type="checkbox"/> <a href="#">id, net</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, controller</a> <input checked="" type="checkbox"/> <a href="#">id, controller, ad_domain</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, endpoint_id, ad_forest</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, active_directory</a> <input checked="" type="checkbox"/> <a href="#">pi, active, pi_id</a> <input checked="" type="checkbox"/> <a href="#">id, transaction</a> <input checked="" type="checkbox"/> <a href="#">model_id, app_id, model, model</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, model, model, proc</a> <input checked="" type="checkbox"/> <a href="#">act, element, activity</a> <input checked="" type="checkbox"/> <a href="#">act, activity, variable</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, app_id, domain</a> <input checked="" type="checkbox"/> <a href="#">model_id, id, directory, ad_forest</a> <input checked="" type="checkbox"/> <a href="#">model_id, endpoint_id, service_id</a> <input checked="" type="checkbox"/> <a href="#">id, app_field_id, creator</a>				
<a href="#">Edit Table</a>   <a href="#">Schema map</a>   <a href="#">Delete all records</a>					

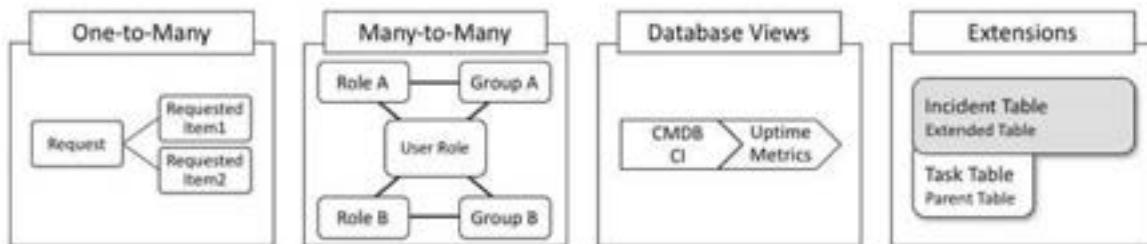
The **Tables & Columns** module provides a view-only list of all existing tables, with columns (fields), column (field) attributes, and indexes.

Navigate to **System Definition > Dictionary** to view a complete listing of records for all tables and table fields in the database.

## Table Relationships

servicenow

Tables can be related to each other in various ways



**One-to-Many:** Within a table, a field can hold a reference to a record on another table. There are three one-to-many relationship fields:

- Reference Fields** - Allows a user to select a record on a table defined by the reference field. Example: Caller field on the Incident table allows a user to select any record on User table.
- Glide List** - Allows a user to select multiple records on a table defined by the glide list. Example: The Watchlist field on the Incident table allows the user to select any record or records on the User table.
- Document ID Fields** - Allows a user to select a record on any table in the instance. Example: Document field on the Translated Text table.

**Many-to-Many:** Two or more tables can be related in a bi-directional relationship, so that the related records are visible from both tables in a related list. Example: software vendors can sell multiple products and products can be sold by multiple vendors.

**Database Views:** Two tables can be joined virtually using the Database Views Plugin to allow for reporting on data that might be stored in more than one table. Database Views are read-only. Create Database Views by navigating to **System Definition > Database Views**.

**Extensions:** A table can extend another table. The extended table includes unique fields plus all of the fields and their properties from the parent table.

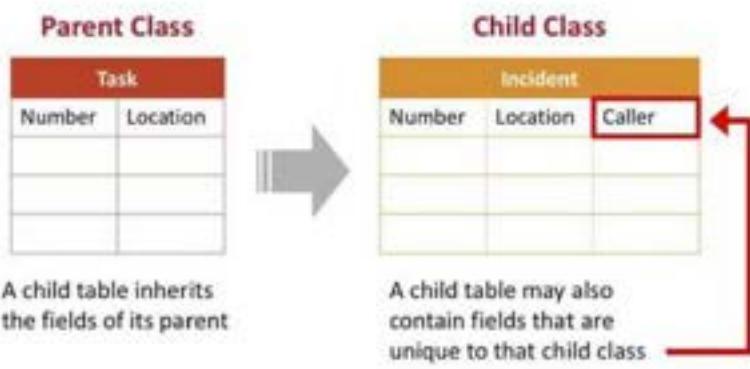
## Extended Tables

service<sup>now</sup>

Tables can extend other tables, creating parent and child tables

A table that extends another table is a **child class**

The table it extends is the **parent class**



In ServiceNow, you can create a new table that stands alone or that extends another table.

The Task table and Configuration table are examples of parent classes that are extended to child classes.

For example, child tables extended from the Task table include Change Request, Incident, and Problem. Child tables extended from the Configuration table include Database, Hardware, and Software.

Extending a table incorporates all of the fields of the original table and allows for unique fields to be created on the new table. The inheritance of the fields of the original table is used to create subcategories of data. Examples include the Incident, Problem, and Change Request tables, which are all subcategories of the **Task [task]** table.

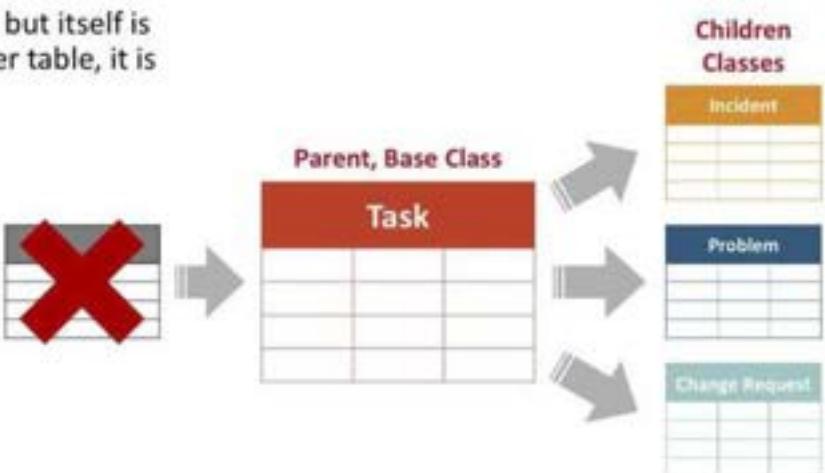
Using the **Dictionary overrides** feature provides the ability to define a field on an extended table differently from the field on the parent table. Examples include overriding the default values, field dependencies, or read-only status of a field.

## Base Tables

servicenow

If a table is extended but itself is not extending another table, it is called a **base table**

The task table is such a table, making it both a parent and a base class



If a field is on a base parent table such as the **Task [task]** table, for example, a different label can be defined for each extended table, such as Incident or Problem. To add a different label for an extended table, navigate to **System Definition > Language File**, then create a new entry for the extended table.

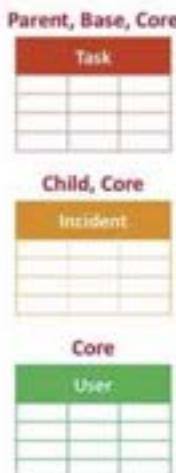
Every child table is a specialization of the previous base table or previous child table. The **Task [task]** table provides a series of standard fields used on every table that extends it.

To extend a table, select the table to extend in the Extends Table field on the table record.

**NOTE:** This option is available only when you are creating a table but not all tables are extensible.

## Core vs. Custom

servicenow



Tables that exist in the base system are called **core tables**

Tables that are created and do not exist in the base system are **custom tables**



Although custom tables are not in the base system, they can still interact with existing core tables.

For example, a reference field on a custom table can access data stored on a core table. By doing so, a relationship between the tables is created which makes them related tables. This relationship is not exclusive between just a custom table and a core table. Related tables can be a combination of multiple core tables and/or multiple custom tables.

## Schema Map

servicenow

The **Schema Map** provides a graphical representation of other tables related to a specific table, either through class extension or reference



In this example map, you will see the Task table as the focus of the map (highlighted in yellow).

Tables with blue bars, including Problem and Change Phase, are tables that **extend** the Task table.

Demonstrated with the Problem table, you can use the Schema Map to identify what columns (fields) originate on Problem, and which columns are inherited from the Task table. Additionally, you can see what field type they are.

Tables with red bars, including Location and User, are tables that are **referenced** by the Task table.

A series of filters at the top of the Schema Map allow you to show/hide tables based on criteria such as whether they are referenced by the Task table, reference the Task table, are extended by the Task table, or extend the Task table.

The **Tables** window on the far right of the screen provides a summary of all the tables presented and their relationships.

### Section Summary

- Database Structure and Hierarchy
- System Definition Table Modules
- Table Relationships
- Schema Map



#### Lab 3.1 – Data Schema:

- Create a new table
- Configure the table form view
- Update the application menu and create new modules

# Lab Topics

- Create a new table
- Configure the table form view
- Update the application menu and create new modules

With an active procedure of testing Infinity devices, Cloud Dimensions needs a method for managing inventory – tracking how many devices have been issued and to whom.

The various teams involved with Infinity testing have come up with a solution but will need the help of the system administrator to implement it.

Their plan is to have this information accessed through an application menu with a series of modules, and be organized into two separate groupings: internal (employee) testers and external (partner) testers.

The primary data point being tracked will be the Infinity devices but information about the users, such as name and email, will also be available.

## Lab 3.1 Data Schema

### Create a New Table

1. As the **System Administrator**, navigate to **System Definition > Tables & Columns**.

The Tables & Columns module provides a clean interface for browsing a list of existing tables in the database. Selecting a table name will display its contents: columns (fields). At that point, you can continue to gather additional information about the field by selecting its name to view some attributes.

This interface provides an easy way to navigate between multiple tables, without having to open individual records to see and compare table content. Please note: all data displayed on this page is read only – it may not be edited, unless you select a table and click the **Edit Table** button.

2. At the top of the **Table Names** list, click **Create Table**:



3. Fill out the top of the Table form with the following changes:

Label: **Infinity**

Name: **u\_cmdb\_ci\_infinity** (auto fills with u\_infinity)

Extends table: **Hardware [cmdb\_ci\_hardware]**

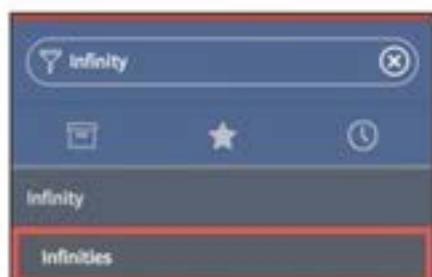
New menu name: **Infinity** (auto fills)

**NOTE:** The Name field automatically populates with u\_infinity. The table name can be changed, as long as it starts with the u\_ prefix indicating it is a custom table. It is best practice to rename the table to indicate it is a custom CMDB CI table.

4. Submit.

### Add Fields to the Infinity Form

1. In the Application Navigator filter field, navigate to the **Infinity > Infinities** module:

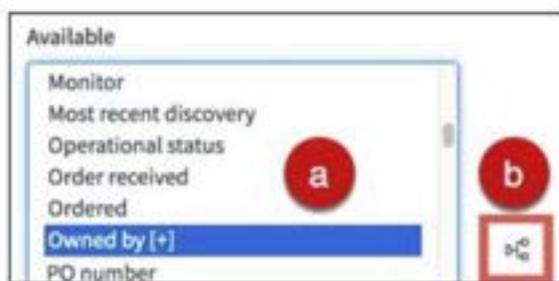


2. An empty List (No records to display) is displayed with default fields.

**NOTE:** In a future lab, you modify the Infinities list layout then import device records from spreadsheets.

3. Click **New** to open a form displaying default fields.
4. Open the **Form Control Menu**, select **Configure > Form Layout**.
5. Remove the **Assigned to**, **Category** and **Fault count** fields, keeping **Name**, **Asset tag**, **Installed**, and **Status** in the Selected list.

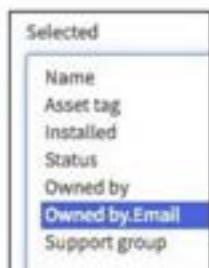
6. Add the **Owned by** and **Support group** fields to the **Selected** list from the **Available** list.
7. Using dot-walking, add **Owned by.Email** to the **Selected** list:
  - a) From the Available list, locate and select **Owned by**
  - b) Click the **Expand selected reference field** icon



- c) From the Available list, scroll down and select **Email**
- d) Select **Add**



You should now see **Owned by.Email** under the **Selected** list:



**NOTE:** This field's value represents the email of the device's owner, and will automatically populate when a value is entered into the **Owned by** field.

8. Repeat the dot-walking process to add the **Owned by.Company** field.

9. Next, in the **Create new field** section, enter:

Name: **Device Number**  
Type: **String** (autofills)  
Field length: **Small (40)** (autofills)

10. Click **Add**.

11. Add two additional fields:

Name	Type	Field Length/Reference
Device Version	String	Small (40)
Device Notes	Journal	

12. After adding the new fields, rearrange the fields under the **Selected** list to look like this:



13. Click the **Save** button.

14. Close the **Saving Form Section** window.

15. The **Infinity** New record form displays:

The screenshot shows a software application window titled "Infinity New record". The form contains the following fields:

- Device Number: [Input Field]
- Name: [Input Field]
- Asset tag: [Input Field]
- Device version: [Input Field]
- Support group: [Input Field]
- Installed: [Input Field]
- Status: Installed [Input Field]
- Owned by: [Input Field]
- Email: [Input Field]
- Company: [Input Field]
- Device notes: [Input Field]

### Update the Infinity Application Menu

The new **Infinity** application menu and its modules will be used to create new **Infinity** device records from the form you have just designed. The list will be used to view **Infinity** device inventory based on employee or partner owners.

1. In the Application Navigator filter field, type **Infinity**.
2. Hover your cursor over **Infinity** and click the **Edit Application** icon (pencil):



3. This brings up the Application Menu record for **Infinity**.

**NOTE:** You could alternately access this record by navigating to **System Definition > Application Menus** and searching for **Infinity** in the list.

4. Update the **Title** to **Infinity Inventory**.
5. **Save** the record.
6. From the **Modules** section, open the **Infinites** record.
7. Update the record as follows:

**Title: All Devices**  
**Order: 200**

8. Click **Update**.
9. Next, click **New** from the Modules section:

The screenshot shows a top navigation bar with 'Modules', 'New' (highlighted in red), 'Go to', 'Order', and 'Search'. Below this is a search bar with a magnifying glass icon and the text 'Application menu + Infinity Inventory'. A table follows with columns: '@', 'Title', 'Table', and 'Active'. A single row is shown with a checkbox, an info icon, the title 'All Devices', the table name 'Infinity [u\_cmdb\_ci\_infinity]', and the value 'true'.

10. Fill out the form:

**Title: Add Inventory**

**Order: 100**

11. Click the **Link Type\*** tab and fill out the fields as shown:

**Link type: New Record**

**Table: Infinity [u\_cmdb\_ci\_infinity]**

**NOTE:** The **Table** field appears to be read-only (indicated by gray), but clicking on the table name will open the drop-down menu.

12. Click **Submit**.

13. Add another module by clicking **New**.

14. Fill out the form as follows:

**Title: Employees**

**Order: 300**

**Link type: List of Records**

**Table: Infinity [u\_cmdb\_ci\_infinity]**

**Filter: Device Number | starts with | CDE**

15. Open the **Form Control Menu**, then select **Save**.

16. Update the form with these changes:

**Title: Partners**

**Order: 400**

**Filter: Device Number | starts with | PAR**

17. Open the **Form Control Menu**, then select **Insert**.

18. Verify that you now have four modules in the Application Navigator:



### Class Discussion

Before creating any table in ServiceNow, the question to always start with is: should the table be created from scratch or by extending an existing table?

**Why was the Configuration Item table extended instead of creating a new table?**

### LAB VERIFICATION

#### New Table - Application Menu and Modules:



#### Add Inventory Module (Form View):

A screenshot of the ServiceNow "Add Inventory" module in Form View. The form contains the following fields:

Device Number	
Name	
Accounting	
Status Number	
Support group	(dropdown)
Model	(dropdown)
Status	(dropdown)
Serial No	(dropdown)
Comments	
Owner	
Location	
Company	
Serial Notes	

*Congratulations, you have completed the Data Schema lab!*

## Module 3

service*now*

### 3.2 CMDB

- What is a CMDB?
- Configuration Items (CIs)
- Relationships and Dependency Views
- Using the CMDB
- Implementation Considerations

## The CMDB and Configuration Items

service<sup>now</sup>

The Configuration Management Database (CMDB) Application is a series of tables and fields that contain all of the Configuration Items (CIs) controlled by your company, as well as their attributes and relationships.

CIs can be tangible or intangible devices or applications (such as firewalls, email servers, etc.) in the Configurations Management Database

Computers



Devices on the network



Applications



Business services



The Configuration Management Database (CMDB) Application provides core functionality for the Configuration Management Database, including modules for hardware and configuration items. This functionality is part of the CMDB plugin, which is activated in a base install.

ServiceNow provides a logical model of your company infrastructure by identifying, controlling, maintaining, and verifying the Configuration Items (CIs) that exist.

ServiceNow's CMDB, in contrast to a static list, tracks not only the Configuration Items (CIs) within your platform, but also the relationships between those items.

The two key CMDB tables are Configuration Item [cmdb\_ci] and CI Relationship [cmdb\_rel\_ci].

A Configuration Item (CI) is any component that needs to be managed in order to deliver Services. CIs typically include business services and their underlying components, such as business applications and hardware.

Configuration Item (CI) Examples:

- Computers
- Devices on the network
- Business services

Click the reference icon to the right of the Configuration item field to be redirected to the selected CIs record in ServiceNow.

A CI record contains all of the relevant attribute data about an item such as name, version, descriptions, ownership, etc., which are documented in fields on the form.

Toggle between the CI Health Dashboard view and the Form view using the options in the form title bar.

The form also contains information about the relationships between CIs. Search for CIs, add new relationships, view the CI Map, or adjust the relationship view settings using the options in the Related Items toolbar.

ServiceNow relationship rules use separate tables to define the relationships between specific CI **base classes** and **dependent classes**. When you extend a table in the CMDB, you must create a new relationship rule in Configuration > Suggested Relationships.

You can view relationships between the current CI and other CIs. An advanced feature is the Related List in CI records which displays additional components contained by that particular CI.

## CI Relationship Editor

servicenow

The **CI relationship editor** uses a concept of suggested relationships to help users see reasonable relationships between configuration items

### Examples:

- A database runs on a server
- A rack provides power for a server

The screenshot shows the CI Relationship Editor interface. At the top, there's a header with the title 'CI Relationship Editor' and the 'servicenow' logo. Below the header, a section titled 'Suggested relationship types' lists several options: 'Application Flow from (Child)...', 'Application Flow To (Parent)...', 'Backup done by (Parent)...', and 'Owner (Child)...'. To the right of this list are three checkboxes: 'Hide CI relationships', 'Hide user relationships', and 'Hide group relationships'. Below the suggested types is a 'Filter' section with two dropdown menus: 'Location' and 'Operational status', each followed by a 'Is anything' dropdown, an 'AND' button, and a 'Run Filter' button. Underneath the filter is a table titled 'Configuration items' with columns: Name, Manufacturer, Location, Description, Client, Updated, and Maintenance schedule. Two rows of data are visible: one for 'Server\_001' located in '432 E Street, San Jose, CA' and another for 'Server\_002' located in '1 Whitehall Court, London'.

Use the CI relationship editor to create CI relationships. Use the CI relationship editor to create CI relationships. It is access from the Related Items toolbar on a Configuration Item form.

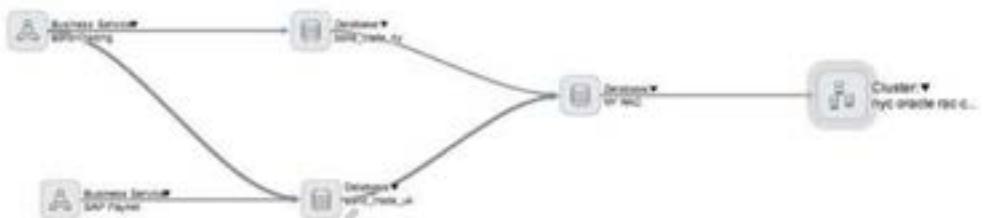
When you use the relationship editor, the CI from which the editor was launched is designated as the base CI. You can then select one or more CIs as a second CI for the relationship. Depending on the selected relationship type, the base CI can become the parent CI or the child CI in the new relationship.

## Configuration Items: Dependency View

servicenow

**Dependency Views** graphically display an infrastructure view for a Configuration item and the business services that it is part of and that it supports

Dependency Views indicate the status of configuration items, and allow access to the CI's related alerts, incidents, problems, changes, and business services

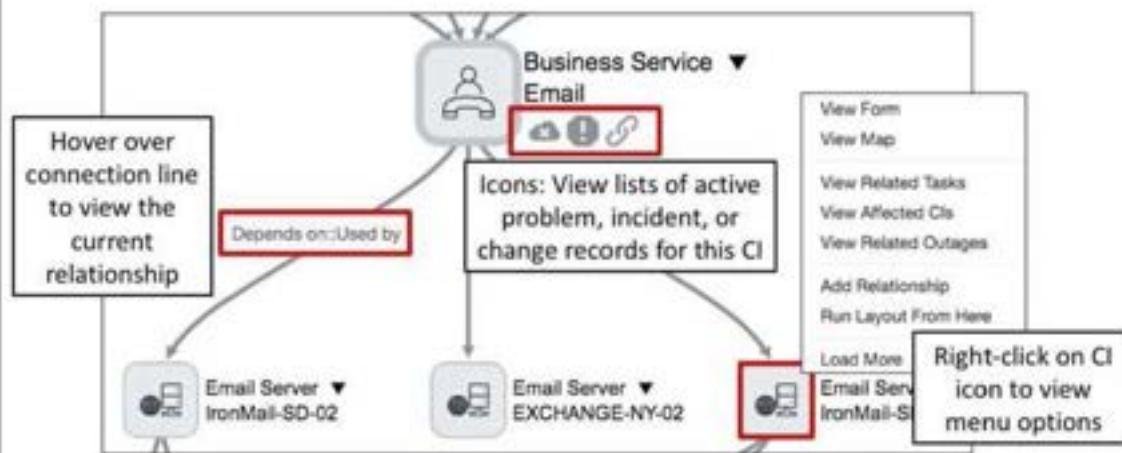


The maps generated by Dependency Views are based on D3 and Angular technology, providing a modern interactive graphical interface to visualize configuration items and their relationships.

Use the Dependency View to view other Configuration Items "upstream" that feed data into an email service, for example, and then "downstream," where you can view all of the other Configuration Items (CIs) that the email service is dependent on.

## Dependency Views: Map Icons

servicenow

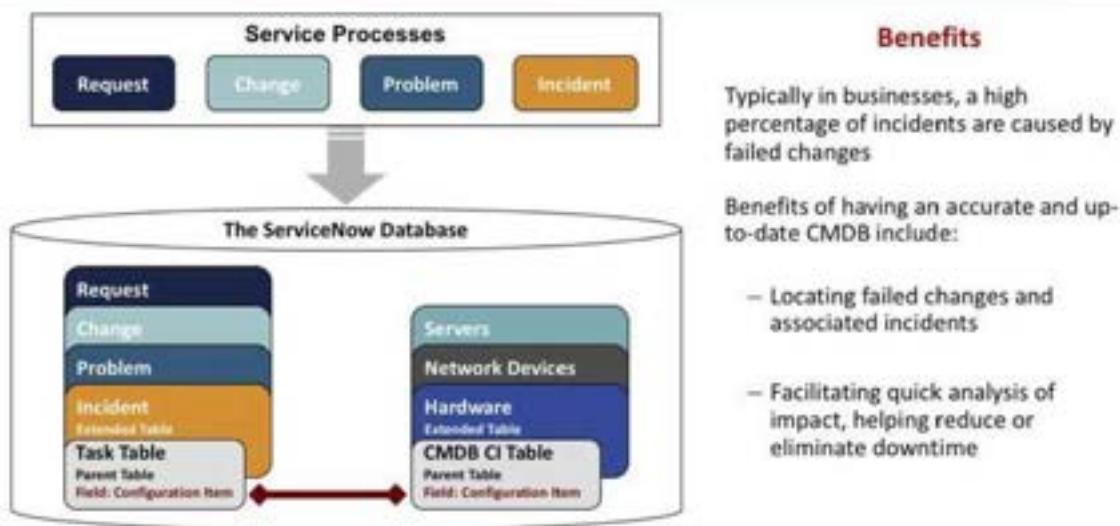


In a Dependency View Map, icons and glyphs indicate whether a CI has an active, pending issue. You can investigate the tasks that are connected to a CI to get more details. The map collapses and expands clusters to make them easier to view.

Understanding the dependencies and other relationships among the CIs will enhance the operational delivery of incident, change, and problem management processes.

## Service Processes: Using the CMDB

service<sup>now</sup>



The Configuration Management Database (CMDB) is a repository of information related to all of the components of an information system. Although repositories similar to CMDBs have been used for years in IT, the origin of the CMDB stems from the Information Technology Infrastructure Library (ITIL). CMDBs help monitor and discover what system components are needed for effective and efficient business processes and IT service management.

All Service Management Processes relate to and involve the CMDB.

For example, someone calls with an issue, and you want to do a root cause analysis, the CMDB gives you insights to effectively troubleshoot.

### Questions to consider before implementing a CMDB:

1. How is data entered or imported then managed?  
Consider people, process, and technology
2. Where is data stored?  
Identify and extend tables accordingly
3. What data is necessary?  
Store only the data being used and omit unused data fields
4. When should imports or refreshes of CI data happen?  
Schedule imports to maintain up-to-date data

A core component in IT best practices, the CMDB serves to maximize the alignment of human capital, technology, and business processes. Setting up a CMDB is a company commitment with measurable financial benefits once everything is set up. You need to keep data current and plan refreshes.

Review the base instance tables when implementing the CMDB. A table you want to create may already exist. Also, prepare and draw out your CMDB schema beforehand so you know what tables you are going to use, which ones you are going to turn off, and what relationships you are going to allow.

Methods for populating the CMDB include: Import Sets, integrating with external CMDBs, and manual input.

Additionally, tools in ServiceNow such as **Help the Help Desk** and **Discovery**, a licensed offering, are efficient methods for gathering configuration item data.

### Section Summary

- What is a CMDB?
- Configuration Items
- Relationships and Dependency Views
- Using the CMDB
- Implementation Considerations



#### Lab 3.2 – CMDB:

- Create a new CI
- Define CI relationships

**Lab Dependency** – This lab requires the completion of Lab 3.1.

# Lab Topics

- Create a new CI Class
- Define CI relationships

**Lab Dependency** – This lab requires the completion of Lab 3.1.

All Infinity devices rely upon a media server, to process and stream data, which will require tracking in the CMDB.

The media server can be categorized under the existing CI Server Class.

With both the Infinity and media server being tracked in the CMDB, a relationship between the two CI Classes will be established – this relationship could be used by Cloud Dimensions to potentially identify an impact of a change management request or outage.

## Lab 3.2 CMDB

### Explore the CI Class Manager

To begin, explore the CI Class Manager interface. The CI Class Manager displays the entire CI Class hierarchy in a tree-view format, consolidating class definitions into a central location. It enables an easier method for viewing, modifying, or extending CI Classes.

A CI Class represents a type of Configuration Item or essentially a table collecting certain data, such as Applications, Computers, Printers, Servers, etc.

In the last lab, a new CI Class was defined for the Infinity – categorizing it as a Hardware CI because of the cmdb\_ci.hardware table extension.

1. Configuration > CI Class Manager.

- From the **Class Hierarchy** panel, scroll down to locate and expand the **Hardware** section, to find the **Infinity** class:

The screenshot shows two panels. On the left is the 'Class Hierarchy' panel, which is a tree view of various ServiceNow classes. A red box highlights the 'Hardware' node under the 'Computer' category. Under 'Hardware', the 'Infinity' class is also highlighted with a red box. On the right is a 'Table Configuration Item' form for the 'infinity\_ci' record. The form includes fields for 'Label' (set to 'Configuration item') and 'Name' (set to 'infinity\_ci'). It also shows a 'Relationships table' with one entry: 'Base Configuration item' (Asset, Reference, Asset). The 'Fields' tab is selected in the top right.

**NOTE:** This confirms that the **Infinity** table and its consequential data records are tracked in the CMDB even though they are accessed from their own application menu and respective modules.

### Add a New Infinity Device to the CMDB

- Infinity > Add Inventory.**

**NOTE:** Unlike existing CI Classes, the **CI Relationships** section is not visible on the new **Infinity** CI Class. The form needs to be configured to display this section.

- From the **Form Control Menu** select **Configure > Form Layout**.
- Scroll down the list of Available fields to locate **CI Relations**:

The screenshot shows a list of available fields under the heading 'Available'. One of the fields, 'CI Relations', is highlighted with a red box. Other visible fields include 'Vendor', 'Warranty expiration', and 'CMDB Baseline Diff'.

This field displays after the alphabetically sorted list of available fields, as it is a formatter rather than a standard field. It will not display on the form layout until the record has been saved.

4. Add the **CI Relations** field to the bottom of the Selected list.
5. Click the **Save** button.
6. Fill out the Infinity New Record form as shown:

Device Number: **CDE0100100**  
Name: **Infinity Testing Device**  
Support Group: **Infinity Support**  
Installed: [today's current date and time]  
Owned by: **Kevin Edd**  
Email: **kevin.edd@example.com** (auto-fills)  
Company: **Cloud Dimensions** (auto-fills)

7. **Save**.

**NOTE:** The Related Items (CI Relations) section now appears.

### Create a New Suggested CI Relationship

We will create a new suggested CI relationship between Infinity and the Infinity Media Server. The Infinity Media Server is responsible for sending content to Infinity devices.

With a CI relationship defined, tools like the CI dependency view can be used by Infinity Support agents to identify the level of impact when issues occur.

1. **Configuration > Relationships > Suggested Relationships**.
2. Click **New**.
3. Fill out the form as shown:

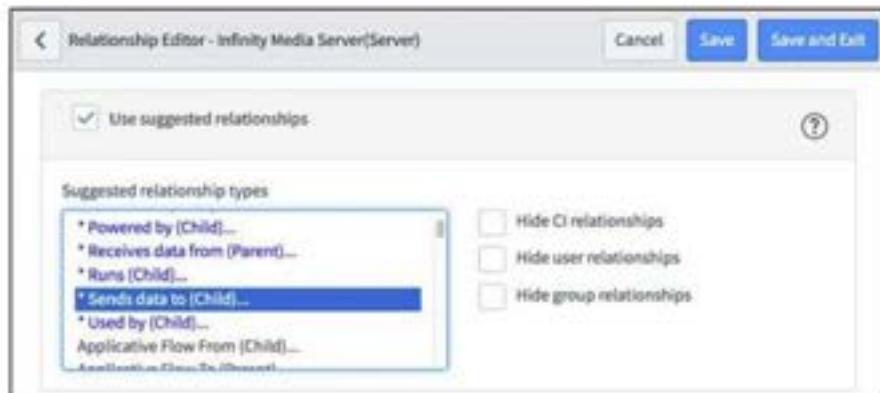
Base class: **Infinity**  
Relationship: **Receives data from (parent)**  
Dependent class: **Server**
4. **Submit**.
5. **Configuration > Servers > All**.
6. **New**.
7. Name: **Infinity Media Server**.

8. Save.

9. Scroll down to the Related Items section, then click the **Add CI relationship** icon:



10. In the Suggested relationship types field, select **\* Sends data to (Child)...**



11. From the Configuration Items section, select **Infinity Testing Device** with the checkbox:



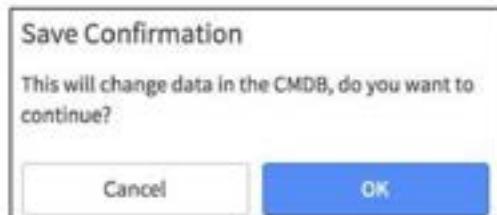
**NOTE:** While only the **Sends data to (Child)** relationship was created, suggested relationships also accounts for the converse relationship, **Receives data from (Parent)**.

12. From the Relationships section, add **Infinity Media Server** by selecting the **Create new relationships** icon (+):



13. Click the **Save and Exit** button.

14. A Save Confirmation pop-up message may display, click **OK**:



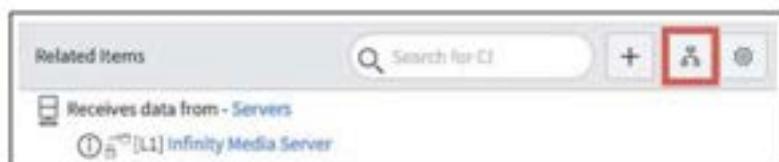
## LAB VERIFICATION

### New Infinity Media Server CI and Relationships

1. Infinity Inventory > All Devices.
2. Locate and open the **Infinity Testing Device** record.
3. Verify the Related Items section looks like this:



4. Open the Dependency View by clicking the **Show dependency views** icon from the Related Items toolbar:



5. The Infinity Testing Device Dependency View displays in a new browser tab/window:



**NOTE:** This displays an example relationship between the Infinity Device and Infinity Media Server CI Classes.

*Congratulations, you have completed the CMDB lab!*

## Module 3

service*now*

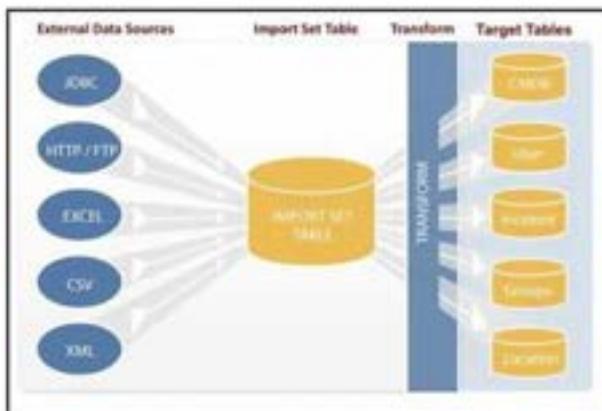
### 3.3 Import Sets

- What are Import Sets?
- Importing Best Practices
- Transform Maps
- Coalesce Fields

## What are Import Sets?

service<sup>now</sup>

An Import Set is a tool used to import data from various data sources, and map that data into ServiceNow tables



Import Sets provide a mechanism to pull data into ServiceNow. Import Sets store data in Import Set tables. Any user logged in with the **admin** or **import\_admin** role can manage all aspects of Import Sets.

**Data Sources** are records in ServiceNow that contain information regarding an Import Set data source. You can import a file from a local source or from a network server by providing a path and authentication information. A data source can come from a file, an LDAP connection, or a JDBC connection.

The **Import Set Table** acts as a staging area for records imported from a data source.

**Transform Maps** provide a guide for moving data from Import Set tables to “Target” tables; field mapping provides direct field-to-field data moves.

A Transform Map is a set of field maps that determine the relationships between fields in an Import Set and fields in an existing ServiceNow table (such as Incidents or Users). Once defined, existing Transform Maps can be reused for mapping data from an Import Set to a ServiceNow table. The Transform Map Module enables an administrator to define destinations for imported data on any ServiceNow tables. Transform mapping can be as simple as dragging and dropping to specify linking between source fields on an Import Set table and destination fields on any ServiceNow table.

The **Target Table** is an existing table in where the data will be placed, post-transformation.

## Importing Best Practices

service<sup>now</sup>

Understand what data you are bringing in and where it should be placed

Plan time before an import to verify your data

- Remove obsolete data before your data import
- Inaccurate data takes time to fix after a data import

Before importing any data, it is important to understand what data you are bringing in and where that data should go.

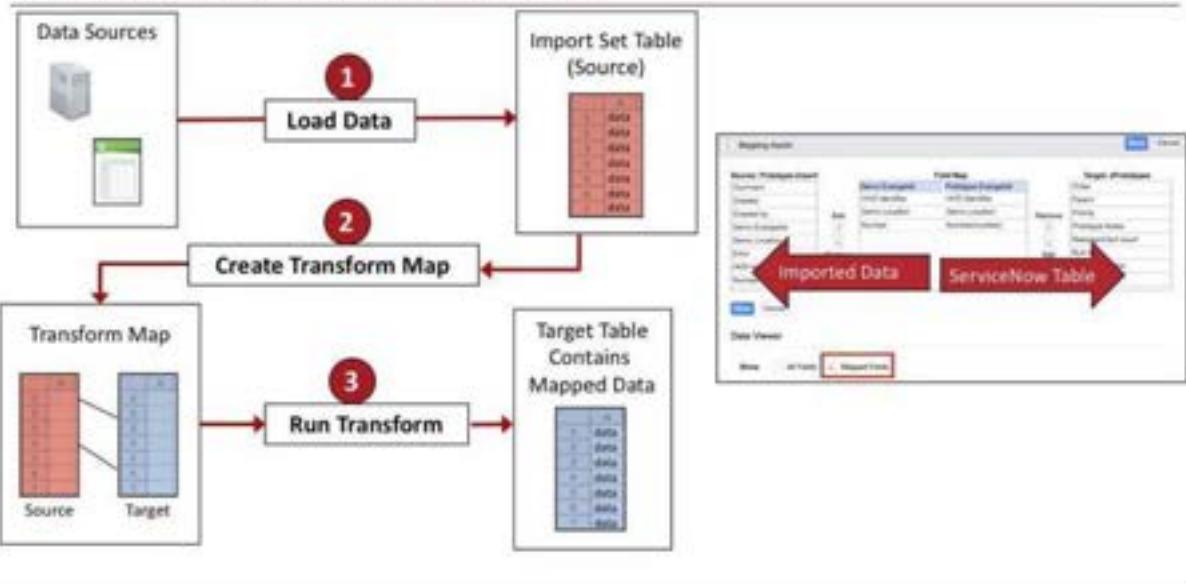
You should verify the data before you import it since bad data will complicate things later in the import and transform processes. Extra time spent planning and examining data before import will save time and potential problems later.

Data should not be imported in extremely large chunks. Creating an extremely large Import Set can cause extensive delays. The imported file label is used to determine the name for the Import Set table that data will be loaded into.

**NOTE:** It is also possible to choose an existing Import Set table to use for loading data from the same source, or data that has the same field/column designations. When an existing Import Set table is chosen, the table fields are added when the incoming source of data contains fields/columns that do not exist.

## Transform Maps and Mapping Assist

service<sup>now</sup>



Transform mapping is flexible; the specification can be as simple as having the application auto-match field names from source and destination, or mapping can use advanced logic and leverage the full power of the ServiceNow scripting environment. A single Import Set field can also be mapped to multiple fields on a Target table. Any table is a potential destination for transformation of an Import Set, and any field within a table can serve as a potential destination for transformation from a field within an Import Set.

Select the table where you want transformed data to be placed. You can select only tables within the currently selected application scope, the global scope, or tables that grant write access to other applications. Name and Source table are set based on the label which was assigned to the Import Set. It is necessary to assign a target table into which the data can be transferred.

**Automatic Mapping Utility:** The simplest mapping method is where all of the field names of the Import Set match the name of the fields on the Target table where the data will be transformed. In this case, simply click Auto Map Matching Fields in the related list in the Table Transform Maps form and confirm proper matching. If there are any discrepancies in terms of how fields were automatically matched, these can easily be corrected using the Mapping Assist utility. When all fields are matched properly, click Transform in the related lists to begin transforming data onto the destination table.

**Mapping Assist Utility:** The Mapping Assist utility provides a visually intuitive environment for specifying mapping between Import Set fields and Target table fields. With the Mapping Assist utility it is possible to map a single source field (field on an Import Set table) to multiple destination fields (fields on a Target table).

## Coalesce Fields Before Data Transform

servicenow

In an import, coalescing on a field (or set of fields) means the field will be used as a unique key

If a match is found using the coalesce field(s), the existing record will be updated with the information being imported

If a match is not found, then a new record will be inserted into the database

There are several possible configurations you can use to coalesce data in Import Sets:

- **No coalesce:** If no coalesce is defined, all imported rows are treated as new records. No existing records are updated. If the import is executed again, duplicate records will be created.
- **Single-field coalesce:** You can coalesce on a single field to update an existing record. If a target table record exists with the same value in the coalesce field as the staging table record, the target table record is updated using the Import Set record values.
- **Multiple-field coalesce:** You can coalesce on multiple fields to update an existing record. If a target table record exists with the same values in all coalesce fields as the staging table record, the target table record is updated using the staging table record values. All coalesce field values between the target and staging tables must match to coalesce with multiple fields.
- **Conditional coalesce:** You can use a script to determine if a staging table row should coalesce to a target record. Most conditional coalesce scripts are defined in the source script field of a field map for the sys\_id field. To update a target record using the staging table record values, the script must return the sys\_id of the target table record.

## Section Summary

- What are Import Sets?
- Importing Best Practices
- Transform Maps
- Coalesce Fields



### Lab 3.3 – Import Sets:

- Modify Infinity All Devices list layout
- Create an Import Set Table and Transform Map
- Clean up the Import Set Table and Transform Map

**Lab Dependency** – This lab requires the completion of Lab 3.1.

# Lab Topics

- Modify Infinity All Devices list layout
- Create an Import Set Table and Transform Map
- Clean up the Import Set Table and Transform Map

**Lab Dependency** – This lab requires the completion of Lab 3.1.

In this lab, you will use Import Sets to load data that has been collected outside of ServiceNow into the Infinity table.

## Lab 3.3 Import Sets

The data will represent asset registration by Cloud Dimensions employees and partners, and include information about the user and their registered Infinity device.

**This lab has three parts:**

1. An initial load, with all of the foundational work required.
2. An incremental load.
3. A data cleanup.

In the **initial load**, you will:

- Gather Excel data files
- Organize a list layout for the Infinity table
- Create a new Import Set by importing data from an Excel spreadsheet to a staging table, then validate the data
- Create a Transform Map based on the staging table; use automapping and mapping assist to establish the mapping between the source and the target tables
- Complete the transform and verify the resulting data

In the **incremental load**, you will:

- Upload a second Excel spreadsheet, reusing the existing staging table and Transform Map
- Identify a key (coalesce) field to ensure that existing records are updated from the new imported data and not duplicated

In the **cleanup**, you will clean up the Import Set Table's data.

## Part 1: Initial Load

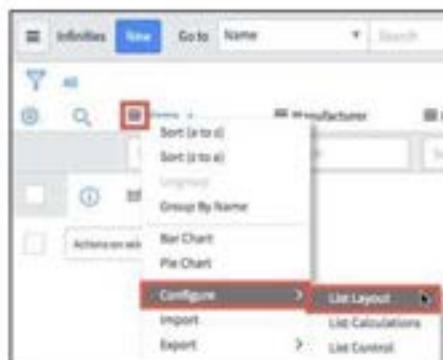
### Gather Excel (.xlsx) Data for Lab

Before you begin, you need to download two Infinity MS Excel files: **infinity-data.xlsx** and **infinity-updates.xlsx**, and save them to your desktop.

Open both the **infinity-data.xlsx** and **infinity-updates.xlsx** files to review their contents: columns, fields, and data types.

### Modify Infinity All Devices List Layout

1. As the System Administrator, navigate to **Infinity Inventory > All Devices**.
2. Open the **Column Control Menu** from any field, then select **Configure > List Layout**:



3. Using the slushbucket, organize the **Selected** field list to include the following fields in the order seen here:

Device Number
Name
Owned by
Location
Status
Support group
Updated

4. Click **Save**.
5. Your All Devices list header should look like this:

Device Number	Name	Owned by	Location	Status	Support group	Updated
1	Test Device	John Doe	Office A	Active	IT Support	2017-01-01

## Create a New Import Set

1. System Import Sets > Load Data.

2. Populate the Import Set form:

Import set table: **Create table** (auto selected)

Label: **Infinity Imports**

Name: **u\_infinity\_imports** (automatically populates)

Source of the import: **File** (auto selected)

File: Choose file, then select **infinity-data.xlsx**

3. Click **Submit**.

4. Review the Progress screen, you should see seven inserts:



## Validate Data in Import Set

1. To verify the data in the new Import Set, in the **Next Steps...** section of the Progress screen, click the **Loaded data** link:



2. Confirm the **Infinity Imports** data loaded correctly:

The screenshot shows a ServiceNow interface for the 'Infinity Inventory Imports' table. The top navigation bar includes 'New', 'Go to Set', and a search bar. Below the header, there are three search fields: 'Row', 'Set', and 'State'. The main area displays 8 rows of data, each with a checkbox, a circled 'i' icon, a row number (1 through 8), a value 'ISET0010001', and a status 'Pending'.

		Row	Set	State
	(i)	9	ISET0010001	Pending
	(i)	3	ISET0010001	Pending
	(i)	3	ISET0010001	Pending
	(i)	1	ISET0010001	Pending
	(i)	2	ISET0010001	Pending
	(i)	4	ISET0010001	Pending
	(i)	2	ISET0010001	Pending
	(i)	6	ISET0010001	Pending

**NOTE:** You may notice a different order.

### Create Transform Map

1. **System Import Sets > Create Transform Map.**
2. Fill out the form as shown:

Name: **Infinity Assets**  
Source table: **Infinity Imports [u\_infinity\_imports]**  
Target table: **Infinity [u\_cmdb\_ci\_infinity]**
3. **Save.**
4. Scroll to Related Links, then click **Auto Map Matching Fields**.

**NOTE:** A verification message displays at the top of your form.

5. Verify that three fields auto mapped, **Name**, **Device Number**, and **Support Group**:

Field Maps (3)		
Field Maps New		
Source Field	Target Field	Coalesce
u_name	name	false
u_device_number	u_device_number	false
u_support_group	support_group	false

**NOTE:** Your field order may be different.

6. In the Related Links list, click **Mapping Assist**.
7. From the Source: Infinity Imports staging table, move **Device Owner** and **Owner Location** into the Field Map.
8. From the Target: Infinity table, move the **Owned by** and **Location** into the Field Map:

Field Map	
Name	Name
Device Number	Device Number
Support Group	Support group
Device Owner	Owned by
Owner Location	Location

9. Click the **Save** button, then verify that there are now five mapped fields:

Field Maps (5)		
Field Maps New		
Source Field	Target Field	Coalesce
u_name	name	false
u_device_number	u_device_number	false
u_device_owner	owned_by	false
u_support_group	support_group	false
u_owner_location	location	false

## Run the Transform

1. In Related Links, click **Transform**.
2. Verify the **Infinity Assets – u\_cmdb\_ci\_infinity** map is selected
3. Click the **Transform** button.
4. The **Progress** screen displays the transformation confirmation messages:



## Verify Infinity Inventory Import

1. **Infinity Inventory > All Devices**.
2. Your screen should show **8 total records**.  
**NOTE:** Infinity record CDE0100100 existed, as it was added in a previous lab.
3. Navigate to **Infinity Inventory > Employees** to confirm seven records display.
4. Navigate to **Infinity Inventory > Partners** to confirm only one record displays.

## Part 2: Incremental Load

### Import Additional Data using an Existing Import Set Table

You will now work with the second spreadsheet you downloaded at the start of this lab: **infinity-updates.xlsx**. For the incremental load, as this spreadsheet is in the same format, you will use the same Import Set Table (Infinity Imports) and Transform Map.

#### Upload the Data

1. **System Import Sets > Load Data**.
2. For **Import set table**, select the **Existing table** radio button.

3. Fill out the form as shown:

Import set table: **Infinity Imports [u\_infinity\_imports]**

Source of the import: **File** (auto selected)

File: Choose file, then select **infinity-updates.xlsx**

4. Click **Submit**.

5. You should see **19 inserts**.

The import of the Infinity data to the staging table is complete, but you are not ready to run the transform yet because you need to add a coalesce. That is, tell the system what the key field is to ensure that existing records are updated rather than adding duplicate records from the imported data.

## Define the Coalesce Field

1. **System Import Sets > Administration > Transform Maps.**

2. Open the **Infinity Assets** Transform Map.

Set the device number (**u\_device\_number**) field as the coalesce, as it contains a unique value. If a match is found for this field, the record will be updated instead of creating a new record.

3. Scroll to the **Field Maps Related List**.

4. In the **u\_device\_number** row, in the **Coalesce** column, double-click the word **false**, then select **true** from the list:

The screenshot shows the 'Field Maps' list page. The 'Coalesce' column for the 'u\_device\_number' row is highlighted with a red box around the 'false' button, indicating it is being modified. A dropdown menu is open over the 'false' button, showing the options 'true' (selected), 'green checkmark', and 'X'.

Source Field	Target Field	Coalesce
u_DATE	name	false
u_device_number	u_device_number	<b>true</b> (highlighted)
u_device_server	owned_by	false

5. Click **Save** (green checkmark) to update the value to true.

**NOTE:** A system message displays at the top of the form.

## Prepare and Run the Transform

1. Under the Related Lists section, click the **Transform** link.
2. In the Selected maps box, verify you are using the correct Transform Map: **Infinity Assets – u\_cmdb\_ci\_infinity**
3. Click the **Transform** button.
4. Verify the transformation complete message.

## Verify Infinity Inventory Import Updates

1. **Infinity Inventory > All Devices**.
2. There are **19 total records** that show.

Notice that some of the records have an update time from the **Initial upload** (**infinity-data.xlsx**) and some have an updated time from the **incremental upload** (**infinity-updates.xlsx**).

Also notice that in the **infinity-updates.xlsx** spreadsheet, in record **CDE0100102**, there was no **Owner Location** information. Compare this to the same record in the **infinity-data.xlsx** spreadsheet, which had Owner Location information.

If you had **Copy empty fields** checked and active in the Transform Map, it would have removed the location data for this record. Since **Copy empty fields** was not active, the data from the original import still remains in the table.

## Part 3: Clean Up Import Set Tables

1. **System Import Sets > Import Set Tables > Cleanup**.
2. Add the **Infinity Imports [u\_infinity\_imports]** table to the **Delete these tables** box.
3. The checkbox for **Delete related transform maps** should be **unchecked**.
4. The checkbox for **Delete data only (preserve table structure)** should be **selected**.

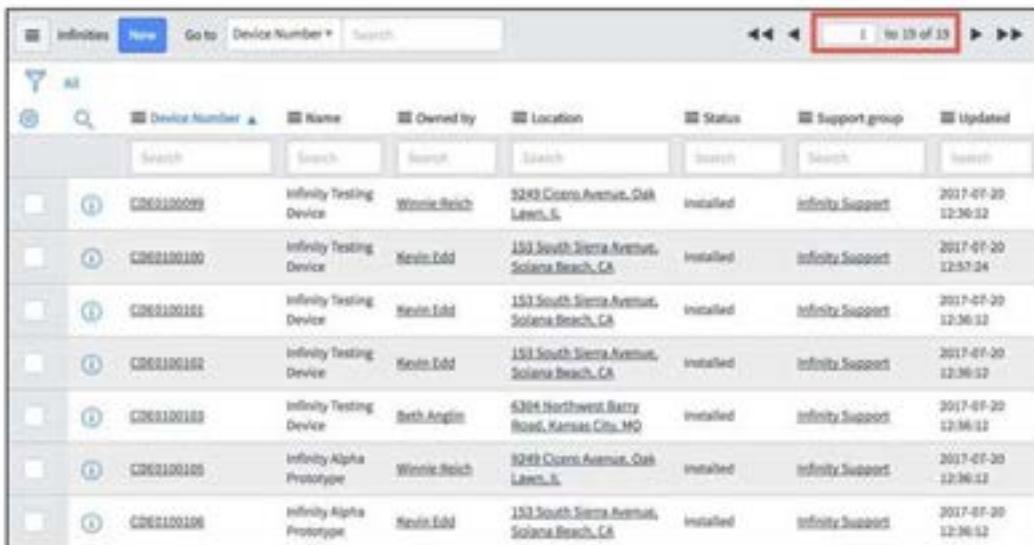
**NOTE:** This will remove the data collected in the Infinity Imports staging table.

5. Click the **Cleanup** button; you should see a Cleanup completed verification message and actions taken displayed in an Import Log.

**NOTE:** If you wanted to delete the Import Set table and any reference to it, including the **Infinity Assets Transform Map**, you would have checked the **Delete related transform maps** checkbox.

## LAB VERIFICATION

### Imported Infinity Inventory Data



A screenshot of a ServiceNow table titled "Imported". The table has columns: Device Number, Name, Owned by, Location, Status, Support group, and Updated. There are 13 rows of data. The first row shows "CDE5100099" as the Device Number, "Infinity Testing Device" as the Name, "Winnie Reich" as the Owned by, "5249 Cisco Avenue, Oak Lawn, IL" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:36:12" as the Updated date. The second row shows "CDE5100100" as the Device Number, "Infinity Testing Device" as the Name, "Kevin Edd" as the Owned by, "153 South Sierra Avenue, Solana Beach, CA" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:37:24" as the Updated date. The third row shows "CDE5100101" as the Device Number, "Infinity Testing Device" as the Name, "Kevin Edd" as the Owned by, "153 South Sierra Avenue, Solana Beach, CA" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:36:12" as the Updated date. The fourth row shows "CDE5100102" as the Device Number, "Infinity Testing Device" as the Name, "Kevin Edd" as the Owned by, "153 South Sierra Avenue, Solana Beach, CA" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:36:12" as the Updated date. The fifth row shows "CDE5100103" as the Device Number, "Infinity Testing Device" as the Name, "Seth Anglin" as the Owned by, "6204 Northwest Barry Road, Kansas City, MO" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:36:12" as the Updated date. The sixth row shows "CDE5100105" as the Device Number, "Infinity Alpha Prototype" as the Name, "Winnie Reich" as the Owned by, "5249 Cisco Avenue, Oak Lawn, IL" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:36:12" as the Updated date. The seventh row shows "CDE5100106" as the Device Number, "Infinity Alpha Prototype" as the Name, "Kevin Edd" as the Owned by, "153 South Sierra Avenue, Solana Beach, CA" as the Location, "Installed" as the Status, "infinity.Support" as the Support group, and "2017-07-20 12:36:12" as the Updated date. The table has search bars at the top for each column and a red box highlights the page number "1 / 10 of 10" in the top right corner.

Device Number	Name	Owned by	Location	Status	Support group	Updated
CDE5100099	Infinity Testing Device	Winnie Reich	5249 Cisco Avenue, Oak Lawn, IL	Installed	infinity.Support	2017-07-20 12:36:12
CDE5100100	Infinity Testing Device	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	infinity.Support	2017-07-20 12:37:24
CDE5100101	Infinity Testing Device	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	infinity.Support	2017-07-20 12:36:12
CDE5100102	Infinity Testing Device	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	infinity.Support	2017-07-20 12:36:12
CDE5100103	Infinity Testing Device	Seth Anglin	6204 Northwest Barry Road, Kansas City, MO	Installed	infinity.Support	2017-07-20 12:36:12
CDE5100105	Infinity Alpha Prototype	Winnie Reich	5249 Cisco Avenue, Oak Lawn, IL	Installed	infinity.Support	2017-07-20 12:36:12
CDE5100106	Infinity Alpha Prototype	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	infinity.Support	2017-07-20 12:36:12

*Congratulations, you have completed the Import Sets lab!*

## Module 3

service*now*

### 3.4 Reporting

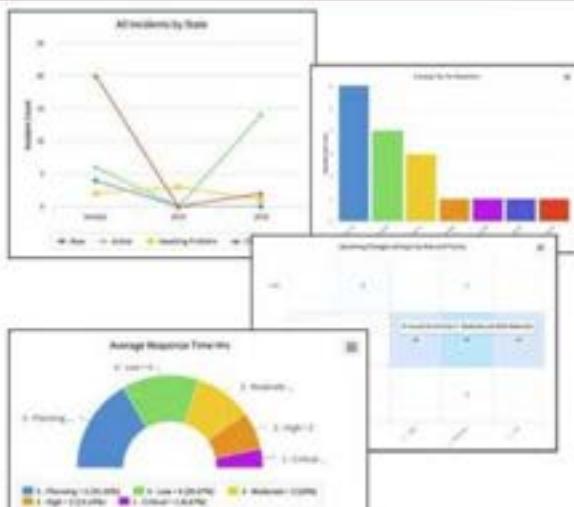
- What is ServiceNow Reporting?
- Report Types
- Creating and Editing Reports
- Report Actions, Options, and Distribution
- Metrics
- Analytics
- Dashboards

## What is ServiceNow Reporting?

service<sup>now</sup>

Use the **Reports** application to run predefined reports and create custom reports that show results, prepared on an ad-hoc basis

Use the Report Designer to create reports by modify an existing report, or create a new report from scratch



The **Reports** application allows you to graphically view and analyze data from your ServiceNow environment. Reports can take many forms, including bar charts, pie charts, dials, lists, pivot tables, donuts, and more. Reports can be run manually or scheduled to be run automatically.

ServiceNow reports are interactive; you can drill down into the report gauges to view and manipulate the underlying data.

There are a range of predefined reports that pertain to applications and features like Incident Management and Service Catalog requests, including Key Performance Indicator (KPI) reports. If none of the predefined reports meet your needs, you can create your own reports by navigating to the **Reports > View/Run** module. Alternatively, you can simply click most column control menus in any list to generate a report directly from the data in that list.

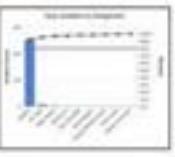
ServiceNow Reporting can easily answer such questions as:

- Did I meet an SLA?
- How many incidents did my team close in a month?
- Metric Reporting: What was the average time from Incident open to Incident closed for each Service Desk team?

For more information, navigate to **Reports > Getting Started**.

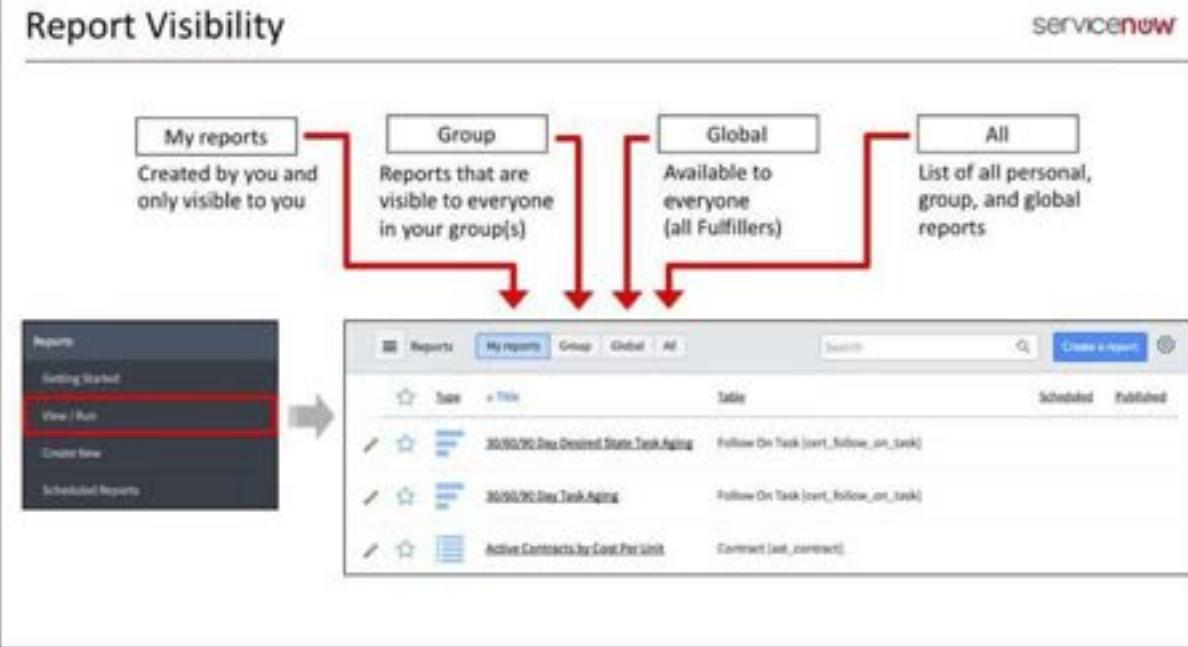
## Report Types

servicenow

<p><b>Scores</b></p> <ul style="list-style-type: none"> <li>– Speedometer</li> <li>– <b>Dial</b></li> <li>– Single Score</li> </ul> 	<p><b>Time Series</b></p> <ul style="list-style-type: none"> <li>– Line</li> <li>– Column</li> <li>– <b>Area</b></li> <li>– Spline</li> </ul> 	<p><b>More</b></p> <ul style="list-style-type: none"> <li>– List</li> <li>– <b>Funnel</b></li> <li>– Calendar</li> <li>– Pyramid</li> <li>– Box</li> <li>– Trend</li> <li>– Control</li> <li>– Trendbox</li> <li>– Map</li> <li>– Pivot Table</li> </ul> 
<p><b>Pies and Donuts</b></p> <ul style="list-style-type: none"> <li>– Pie</li> <li>– <b>Donut</b></li> <li>– Semi donut</li> </ul> 	<p><b>Column and Bar Charts</b></p> <ul style="list-style-type: none"> <li>– Bar</li> <li>– <b>Pareto</b></li> <li>– Histogram</li> <li>– Horizontal Bar</li> </ul> 	
<p><b>Multidimensional</b></p> <ul style="list-style-type: none"> <li>– Bubble</li> <li>– <b>Heatmap</b></li> <li>– Multi-level Pivot Table</li> </ul> 		

The ServiceNow base instance comes with standard report types that have many variations.

For example, a Pareto chart, named after Vilfredo Pareto, is a type of chart that contains both bars and a line graph, where individual values are represented in descending order by bars, and the cumulative total is represented by the line. Pareto Charts are useful to show the significance of factors for a given question/process. Pareto Charts use the rule that about 20% of input produces almost 80% of the outputs.



The **Reports > View / Run** module contains a library of reports which you can run and use to create your own custom reports. Many of these reports came with the platform and others were created by your reporting administrators specifically for your company.

The Reports page contains different sections for reports which are visible to different audiences:

- **My reports**: Visible only to the report creator (Me)
- **Group**: Visible to one or more specific users and/or groups (Groups and Users)
- **Global**: Visible to all users (Everyone)
- **All**: A list of all Global, Group, and personal reports

## Create and Edit Reports

service<sup>now</sup>

### Create New Report

1. Build and run a list filter.

The screenshot shows the ServiceNow Report Designer interface. At the top, there's a toolbar with buttons for 'New', 'Edit', 'Save', 'Print', 'Copy', 'Delete', 'Add Sub...', and '%'. Below the toolbar is a section titled 'List Filter' with the sub-section 'Advanced Group'. It contains two filter conditions: 'Active' set to 'Is' and 'Priority' set to 'Is Not'. To the right of the filters are buttons for 'Run', 'Print', and 'Close'. At the bottom right of the filter area is a red box around the 'Run' button.

2. Open the Column Control Menu, then choose Bar Chart or Pie Chart.

The screenshot shows a report preview with a column control menu open. The menu has sections for 'Cell', 'Priority', 'Type', and 'Category'. Under 'Priority', there are options for 'Sort by Priority', 'Show Visual Task Board', and 'Group by Priority'. In the 'Chart Type' section, 'Bar Chart' is highlighted with a red box. Other options like 'Pie Chart' are also visible.

### Edit Existing Report

Modify an existing report in the Reports Application, navigating to Reports > View / Run

You can build reports from scratch, but it is often easier to start with a filtered list or an existing report. When you start with an existing report, reporting configuration choices will be provided for you. By modifying the provided options, you can configure the report to meet your needs. It is a best practice to copy a base report then edit your copy.

Use the ServiceNow Report Designer to:

- Leverage reporting visibility and available report types
- Use multi-level filters, filter operators, and sort order to refine reports
- View, create, edit, and schedule reports
- Work with reporting roles
- Use Related Tables (dot-walking and Database Views)

Advantages to modifying an existing report:

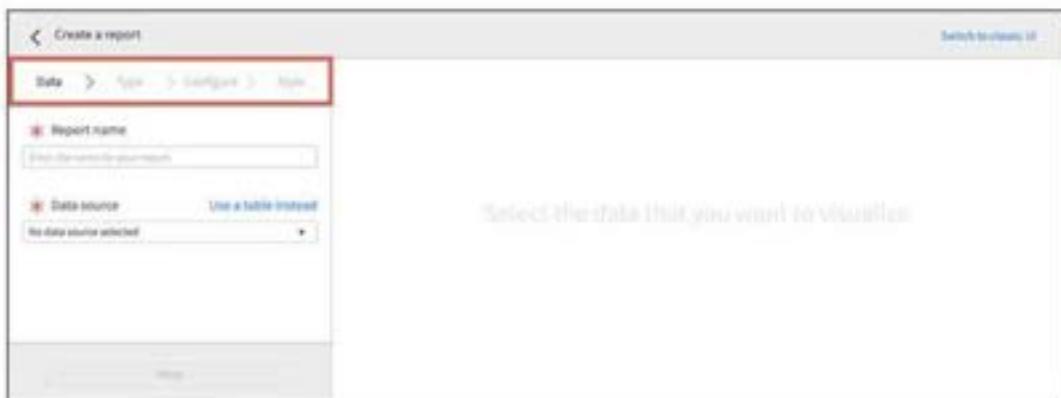
- You can start with a report that already has the basic information and make minor changes to get what you need
- Browsing existing reports helps you learn which of the tables are relevant to the work you do
- Helps you learn different uses for the various report types
- You can leverage ITIL best practices by using Key Performance Indicator (KPI) reports

## Report Designer: Creating a Report

service*now*

**Report Designer** is an interface used for creating or modifying ServiceNow reports

It features four sections which provide reporting configuration options: Data, Type, Configure, and Style



Users with the admin role can access the **Reports > Administration > Properties** module to enable the Report Designer for all users with the permissions to create/edit Reports. Alternatively, users will be able to revert back to the classic reporting interface by clicking on the **Switch to classic UI** link towards the top-right of the Report Designer interface.

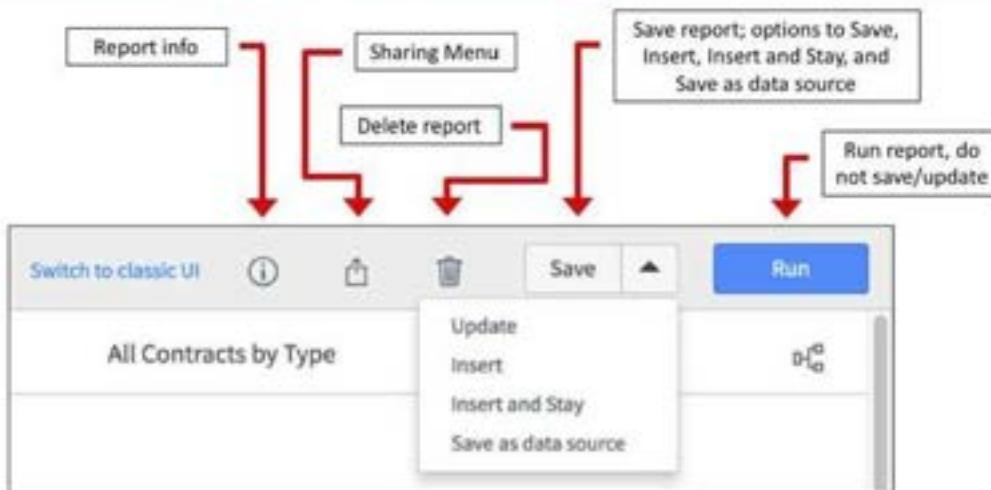
Each section of the Report Designer provides different configuration options:

- **Data:** Provide a name for the report, as well as select the source from where your data comes from. You can choose a data source, which is a predefined data set used for creating reports; or a ServiceNow table.
- **Type:** Select the visualization of your report by choosing a report type. There are 27 different types to choose from!
- **Configure:** Do things like group the data by a specific field(s) and run calculations against the data.
- **Style:** Adjust the look of your report, from coloring to titles, as well as making adjustments to the report legend.

**NOTE:** Every time you make an adjustment through these controls, remember to click the **Run** button in the top-right to redraw the report with your changes.

## Report Designer: Report Actions and Options

servicenow



The Report actions available to you depend on your role. Many of the Report actions are easily understood; therefore only some of them are detailed here:

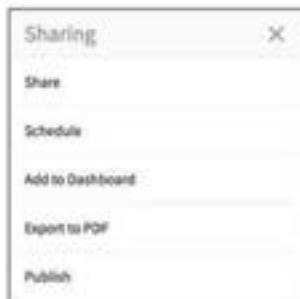
- **Update:** Overwrite report, return to the report list
- **Insert:** Save a duplicate copy of the report, return to the report list
- **Insert and Stay:** Save a duplicate copy of the report, remaining on the report
- **Save as report source:** Allows you to create a pre-defined data set that can be used for creating reports

## Report Designer: Report Distribution

servicenow

### Share

To make a report visible to a particular group or user, use the **Share** option to select Groups and/or Users



### Publish

Steps to publish and view a report:

1. With desired report displayed click the **Sharing** menu icon, then click **Publish**
2. Click the **Copy report link** icon from the report header to copy the URL to your clipboard
3. **Open URL** in browser

From the **Sharing** menu, you have the following options:

- **Share:** Allows you to specify who can see the report. Options include Me, Everyone, and Groups and Users. Admin role is required for Everyone and Group sharing.
- **Schedule:** Create a scheduled email of the report
- **Add to Dashboard:** Add directly to a Dashboard on a homepage you choose, or within a Performance Analytics tab
- **Export to PDF:** Convert the report to a PDF which can be generated immediately or sent as an email
- **Publish:** Create a public URL for this report. Users may need to log into ServiceNow to view the report and have an appropriate role in order to view all of the data.

When distributing a report, sharing has the ability to make the report visible to authenticated users within ServiceNow.

Publishing a report makes it available to users outside of ServiceNow but does not necessarily share the underlying data. Access Control rules restrict visibility to the underlying data but not to all reports. For this reason, it is recommended to be cautious when publishing reports for external visibility.

## Reports and Metrics

service<sup>now</sup>

When you **report** on a table (for example, Incident or Problem), information about the current state of platform data displays

A **metric** is used to measure and evaluate the effectiveness of IT service management processes

- Metrics measures data over time to show past history
- Metrics can gather data as the data is updated

System tables are, by default, restricted from the reporting module. These tables include, but are not limited to the following: syslog, syslog\_transaction, sys\_attachment, and sys\_email.

Compared to reports, **metrics** are used to measure and evaluate the effectiveness of IT service management processes. A metric could measure the effectiveness of the incident resolution process by calculating how long it takes to resolve an incident.

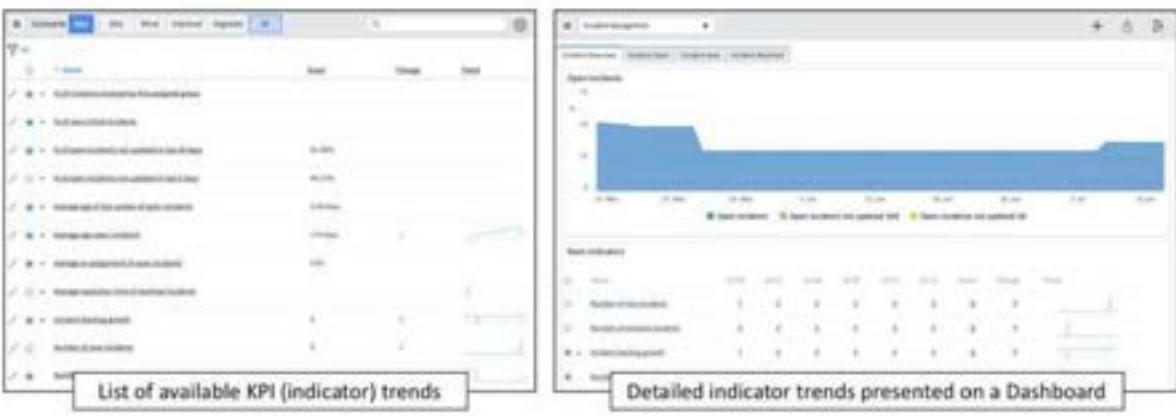
Sometimes a metric can be easily obtained from the data. For example, to find the number of incidents that were created today, a report will count the number of incident records in the Incident table with a created date of today. Other times, metrics need to be gathered as data is updated. For example, determining how long an incident was assigned to a certain group requires collecting information about assignment changes and calculating the duration of each assignment.

The Metric Definition plugin provides a declarative way of defining metrics, and, once defined, the data for the metric is gathered, and occurrences of the metric are calculated and stored.

## Performance Analytics

servicenow

Performance Analytics enables you to track and aggregate data over time through elements called **scorecards** and **indicators**



Performance Analytics enables you to track and aggregate data over time, such as to measure how many tickets are resolved each week per assignment group. Performance Analytics is enabled for the Incident table by default. To track data for other tables and applications, you must license Performance Analytics.

Performance Analytics samples source data on a daily basis to build a trend over time.

Performance Analytics for Incident Management comes with several predefined elements that you can use to assess organizational performance, including:

- **Indicators:** define the metrics to track based on an indicator source, and specifies an aggregation such as to count the number of new tickets
- **Data collection jobs:** automatically collect scores for automated indicators and breakdowns
- **Scorecards:** display scores for a single indicator and allows you to perform detailed analysis of the metric, such as comparing scores over time
- **Dashboards:** contain actionable data visualizations that help you improve your business processes and practices

Performance Analytics for Incident Management is a limited version of Performance Analytics that is included in the base system, enabling you to become familiar with the functionality.

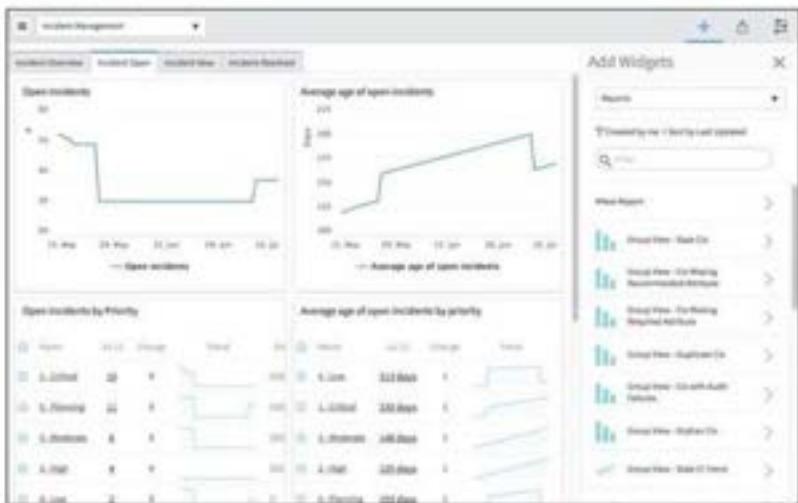
Performance Analytics premium allow you to create indicators and other configuration records such as breakdowns, and collect data for tables other than Incident.

## Dashboards Overview

service<sup>now</sup>

Dashboards enable you to display multiple Performance Analytics, reporting, and other widgets on a single screen

Use dashboards to create a story with data that can be shared



Dashboards may be responsive or non-responsive, but are set as non-responsive by default. Responsive dashboards require the activation of the Performance Analytics plugin on upgraded instances.

With dashboards you are able to:

- Create and edit Performance Analytics reports and other widgets directly from the dashboard
- Use the Add Widgets pane to quickly find and preview widgets, then add them to the dashboard
- Easily share dashboards with other users from the integrated sharing pane
- Use quick layouts to snap widgets into a predefined layout, then adjust the layout as desired
- Set dashboards as your homepage so you can quickly access information that you use frequently

**NOTE:** There is a lot you can do with dashboards so it is encouraged that you to find more information at the ServiceNow product documentation site.

## Section Summary

- What is ServiceNow Reporting?
- Report Types
- Creating and Editing Reports
- Report Actions, Options, and Distribution
- Metrics
- Analytics
- Dashboards

### Lab 3.4

Reporting



Pages 206 – 213



15 minutes

### Lab 3.4 – Reporting:

- Create a report
- Run data collector jobs
- View and modify a Performance Analytics dashboard

**Lab Dependency** – This lab requires the completion of Lab 1.3.

## Lab Topics

- Create a report
- Run data collector jobs
- View and modify a Performance Analytics dashboard

**Lab Dependency** – This lab requires the completion of Lab 1.3.

Stakeholders in groups supporting Infinity would like a report dashboard built so that they may be able to monitor Infinity incident data in real-time and get all of the important information they need on one screen.

Winnie Reich and Audra Cantu have requested this information be made available.

## Lab 3.4 Reporting

### Create a Report

1. Reports > View / Run.

**NOTE:** Although the first part of the lab is completed as the System Administrator, any user with the right permissions can access the **Reports** application menu and modules to create and share reports.

2. Use the search bar at the top-right to search for reports containing **incidents by priority** in their title.
3. Once the results display, click **Incidents by Priority and State** to open the report:

The screenshot shows the 'Reports' application interface. At the top, there are tabs for 'Reports', 'My reports' (which is selected), 'Group', 'Global', and 'All'. A search bar contains the text 'Title contains: Incidents by priority'. Below the search bar is a table with columns: Type, Title, + Table, Scheduled, Published. There are two rows of data:

Type	Title	+ Table	Scheduled	Published
Incident [incident]	Incidents by Priority and State older than 30 Days			
Incident [incident]	Incidents by Priority and State			

The second row, 'Incidents by Priority and State', has a red box drawn around its 'Title' column.

4. The Report Designer loads with the report displaying information:

The screenshot shows the 'Edit report' interface in the ServiceNow Report Designer. On the left, there's a sidebar with tabs for 'Data', 'Type', 'Configure', and 'Style'. The 'Data' tab is selected. Below it, there's a field for 'Report name' containing 'Infinity Incidents by Priority and State'. The main area displays a heatmap titled 'Incidents by Priority and State'. The legend indicates five priority levels: Critical (red), High (orange), Moderate (yellow), Low (light green), and Pending (light blue). The heatmap grid shows numerical values corresponding to the number of incidents for each priority level.

5. Select the Data breadcrumb title from the panel on the left, then update the name of the report to **Infinity Incidents by Priority and State**:

This screenshot shows the 'Edit report' interface again, but with a red box highlighting the 'Data' breadcrumb title in the sidebar. The report name field below it has been updated to 'Infinity Incidents by Priority and State'.

Apply a filter on the data so that it displays only Infinity incidents.

6. From the main reporting area, where the data is displayed, click the Open condition builder icon (funnel):

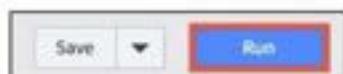
This screenshot shows the main reporting area of the ServiceNow Report Designer. A red box highlights the 'Open condition builder icon' (a funnel icon) located in the top-left corner of the reporting area. The reporting area displays a heatmap titled 'Incidents by Priority and State'.

7. Use the dropdown menus to set a single filter condition:

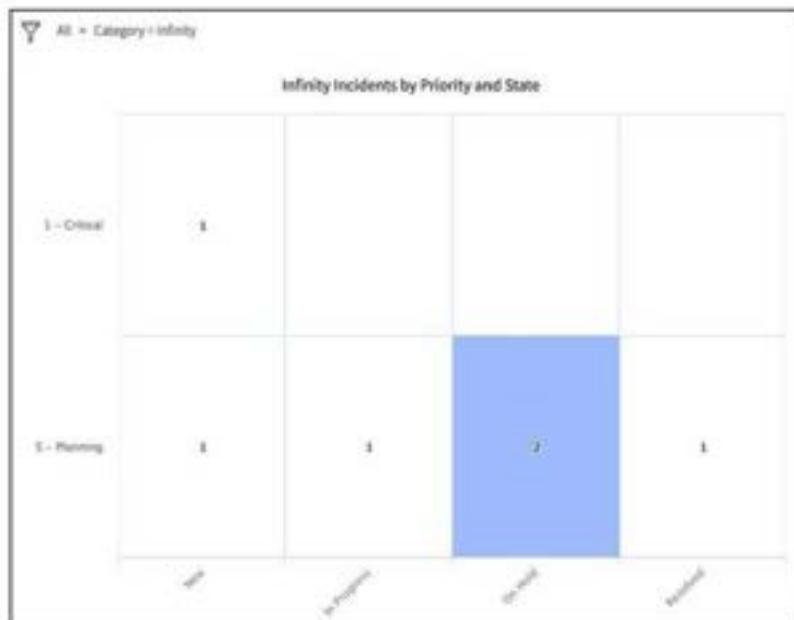
**Category | is | Infinity**



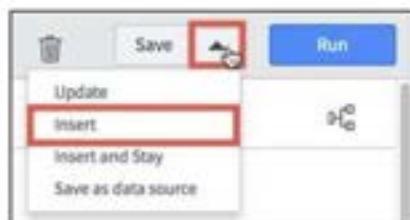
8. From the Report Designer header, click the Run button:



9. Notice the report's data has updated:



10. Open the Save menu by clicking the downward-facing arrow, then select Insert to save a copy of the new report you created:



11. From the Reports list, search for reports containing **Infinity** in their title. Notice there are **three** Infinity incident reports:

The screenshot shows a list of reports. At the top, there are tabs: Reports, My reports, Group, Global, and All (which is selected). Below the tabs, there is a header row with columns for Type, Title, and Table. There are three data rows:

Type	Title	Table
Infinity Incidents by Priority and State	Incident [incident]	
Open Infinity Incidents	Incident [incident]	
Unassigned Infinity Incidents	Incident [incident]	

**NOTE:** These other Infinity reports were created as demo data.

## Run Data Collector Jobs

1. **Performance Analytics > Data Collector > Jobs.**

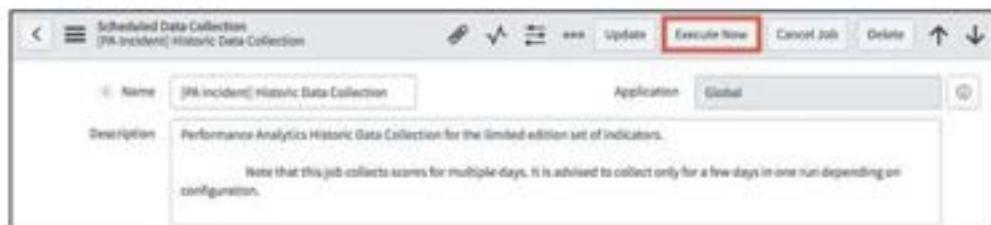
Running data collector jobs allows you to aggregate historic and recent data entered into ServiceNow which can then be used to display on reports and dashboards. Data collection can be executed manually or scheduled to run automatically.

2. From the list, open the job named **[PA Incident] Historic Data Collection**:

The screenshot shows a list of scheduled data collection jobs. At the top, there are buttons for New, Go to, Time, and Search. A filter icon shows 'All' and 'Run Once'. Below the filters, there is a header row with columns for Name and Run. There are five data rows:

Name	Run
[PA Incident] Daily Data Collection	Daily
[PA Incident] Historic Data Collection	On Demand
CMDB Benchmarking Data Collection	Daily
Benchmark Data Collection	Daily

3. After the form loads, click the **Execute Now** button:



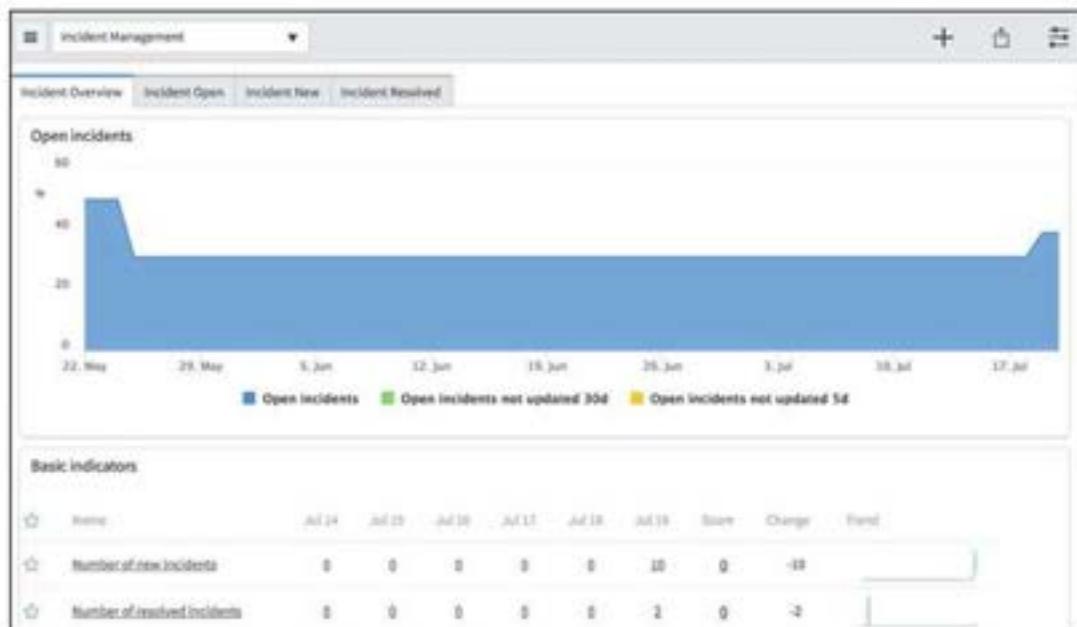
4. Repeat this process for the other job named **[PA Incident] Daily Data Collection**.

### Restrict Access to the Infinity Inventory Application

Now with the data collector jobs ran and data aggregated, let's look at a Performance Analytics dashboard for Incident Management and update it to support Infinity data.

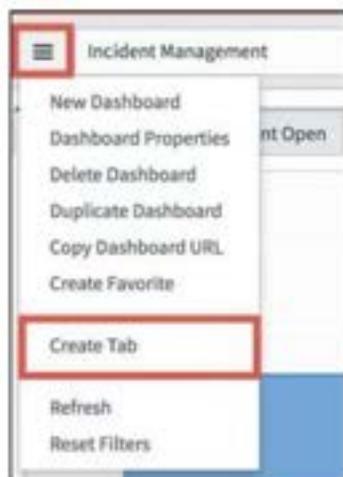
Access to these dashboards requires specific Performance Analytics roles.

1. **Performance Analytics > Dashboards.**
2. The **Incident Management** dashboard loads, which contains four tabs: Incident Overview (currently seen), Incident Open, Incident New, and Incident Resolved:



Click on each of the dashboard tabs to see what data is presented and how.

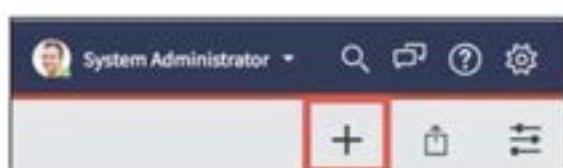
3. Next, add a new dashboard tab by opening the header context menu, then selecting **Create Tab**:



4. Once the new tab appears, hover over it with your cursor, then click the edit icon (pencil):



5. Input the name **Infinity**, then press **Enter** on your keyboard to save the tab name.
6. To add reports to the dashboard, select the **Add Widgets** icon (+) from the dashboard header:



7. Use the drop down menu to find and select **Reports**:



8. Next, type **Infinity** into the Filter search bar to find our Infinity reports.
9. Select the first report, **Infinity Incidents by Priority and State**, to open a preview:



10. Click **Add**.

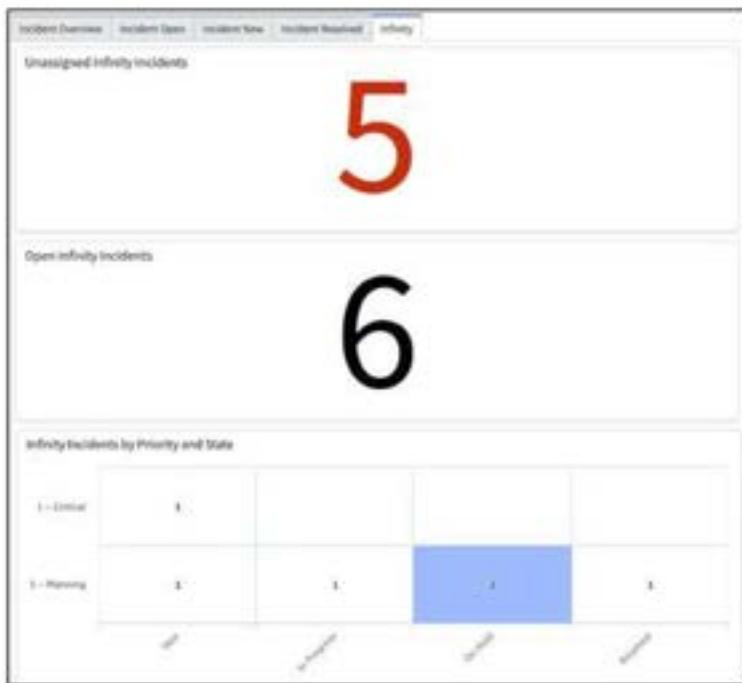
The report is added to the dashboard tab:



11. Return to the widget reports list to add the next report, **Open Infinity Incidents**.
12. Repeat these steps to add the final report, **Unassigned Infinity Incidents**.

## LAB VERIFICATION

### Performance Analytics Dashboard



**NOTE:** You have to scroll down in order to view the contents of all three reports.

***Congratulations, you have completed the Reporting Lab!***

## Module 3

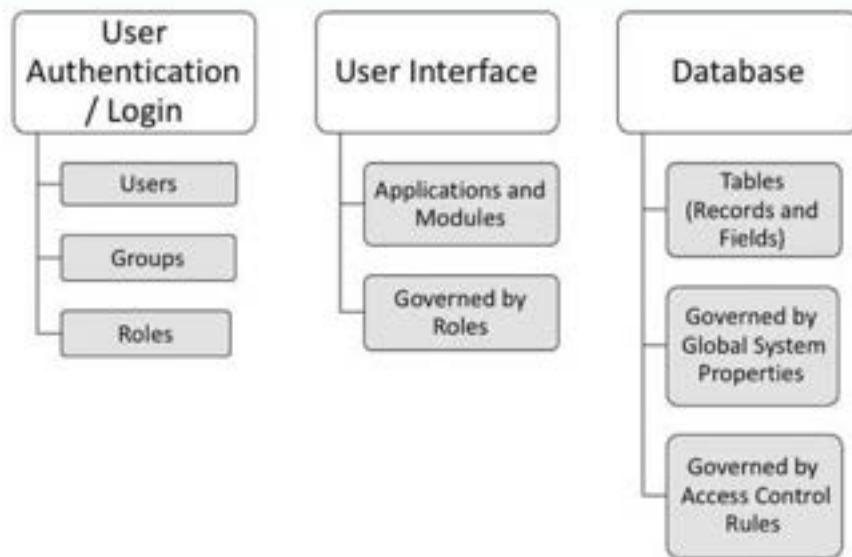
service*now*

### 3.5 Data Security

- User Permissions Summary
- Access Control Rules
  - What is an Access Control?
  - Access Control List (ACL)
  - Rule Types
  - Rule Definition Criteria
  - Using the Wildcard
  - Evaluation Workflows

## Access Overview

service<sup>now</sup>



ServiceNow provides several levels of security before an end user has the capability to perform CRUD (Create, Read, Update, Delete) operations on a table:

- **User Authentication/Login:** Users, Groups, and Roles
- **Application and Modules Access:** Controlled by roles configured at the Application and Module level
- **Database Access:** Access to tables and their records and fields are controlled via globally defined system properties as well as table and field level Access Controls. If a row level rule and a field level rule are in conflict, both rules must be true before an operation is allowed.

There are three security modules typically used by the System Administrator:

- **System Properties > Security**
- **System Security > Access Control (ACL)**
- **System Security > High Security Settings**

## What is an Access Control?

service<sup>now</sup>

An **Access Control** is a security rule defined and set at the row-level (access to the record) and at the column-level (access to the field)

It is executed when attempting to access any ServiceNow table

These rules also restrict CRUD and ServiceNow-specific operations

**C**reate

**R**ead

**U**pdate (write)

**D**elete

Most security settings are implemented using Access Controls.

Access Control rules can restrict CRUD and ServiceNow-specific operations on tables and fields.

ServiceNow operation examples include:

- **execute**: user cannot execute scripts on a record or UI page
- **Edit\_ci\_relations**: user cannot define relationships between Configuration Item [cmdb\_ci] tables
- **Save\_as\_template**: controls the field that should be saved when a template is created
- **Report\_on**: user cannot create reports on the object
- **Personalize\_choices**: user cannot right-click a choice list field and select Configure Choices

## Access Control List (ACL)

servicenow

The **Access Control List (ACL)** contains all of an instance's Access Control rules

Users with the appropriate permissions can modify rules and their definitions

Access Control rules require users to pass a set of requirements in order to gain access to particular data

Each Access Control rule specifies:

- The object being secured
- The permissions required to access the object

All users with the admin role have special access to all platform features, functions, and data because admins can override Access Controls and pass all role checks so grant the admin privilege carefully!

Access Control Rules can be defined in three ways:

1. **Roles:** If you put one or more roles there, then only users with at least one of those roles are allowed to perform the requested operation.
2. **Conditional Expressions:** You will see a condition widget where you can add conditional expressions to your ACL. For example, you might have a conditional expression that says "Active = true" if you want this rule to evaluate to true only for database records that meet a specific condition.
3. **Scripts:** If the Advanced checkbox is selected, you then have an opportunity to apply security based on user defined script. Your script has access to the current record and has responsibility for setting a global "answer" variable to allow, or deny, access to the requested resource/operation. A script is evaluated *in addition* to other conditions you set on the rule.

All must evaluate to true.

## Access Control Types

servicenow

There are 3 types of Access Control rules to identify the object being secured:

1. **table.None**
2. **table.field**
3. **table.\***

①

No specific field selected - this rule applies to the whole table including its records.

②

This rule applies to only one field on a record, in this case the Caller field on an incident record

③

Wildcard – this rule applies to every field on a record without a table.field rule

Each Access Control specifies the table or type of record (including fields), operation being secured, and unique object identifier. Access Control rules are defined for and applied to a specific table so that the rule is within the context of the table and the type of data stored.

## Examples

servicenow

### Rule Example 1

Rule Name: **incident**

Rule Operation: **read**

Definition:

Script

```
1 current.opened_by == gs.getUserID() ||  
current.caller_id == gs.getUserID() ||  
current.watch_list.indexOf(gs.getUserID()) > -1;
```

Description:

You can read an incident if you opened it, are the caller of record, or are on the watch list

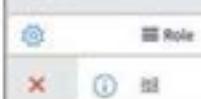
### Rule Example 2

Rule Name: **incident**

Rule Operation: **read**

Definition:

Requires role



Description:

You can read an incident if you have the **itil** role

In the first rule example, the access control's definition includes only a script requirement which roughly translates to: is the user that opened this record the same user attempting to access it, or is the user that is listed as the caller the same user attempting to access it, or is the user that is listed on the record's watch list the same user attempting to access it?

If the user meets any single one of these criteria, they are able to read (or view) the incident record. If they do not meet a single one of these criteria, then they are not able to read the incident record.

To further illustrate this, think of the following scenario: Joe Employee called in to the Service Desk to report an issue. The support agent receiving Joe's call created an incident and added Joe's name to the Caller field. Later, when Joe attempts to view the record – possibly to see any progress made with it – he will have no problem doing so, because of this rule.

Now, while still thinking of this scenario, the agent that opened the incident record would technically be able to view Joe's record because of that same rule. However, what happens if they reassign the incident to another fulfiller user?

Luckily, the second rule example would allow that other fulfiller the chance to view the record. This rule states that in order to read an incident, you have to have the **itil** role.

In the case of these two rules, because they are applied on the same object level (reading an incident record), passing either rule will grant access to the user.

## Using the Wildcard

servicenow

The selection of the wildcard (\*) symbol from the *Name* drop-down list on the Access Control form means that the rule in question applies to all fields on the selected table **except** for those with explicit rule

`change_request.None` read Access Control for admin and itil roles  
`change_request.*` read Access Control for admin role  
`change_request.type` read Access Control for itil role

Admin role is able to see:

#	Number	Short description	Type	Date	Planned start date
1	1000000000	ITSM update	Normal	None	2017-01-01 00:00:00
2	1000000001	CMDB Asset-ITSM Integration Update Asset	Normal	None	2017-01-01 00:00:00
3	1000000002	ITIL	Normal	None	2017-01-01 00:00:00
4	1000000003	Oracle Full Oracle Audit Configuration	Normal	None	2017-01-01 00:00:00
5	1000000004	Java App Remote	Normal	None	2017-01-01 00:00:00

itil role is able to see:

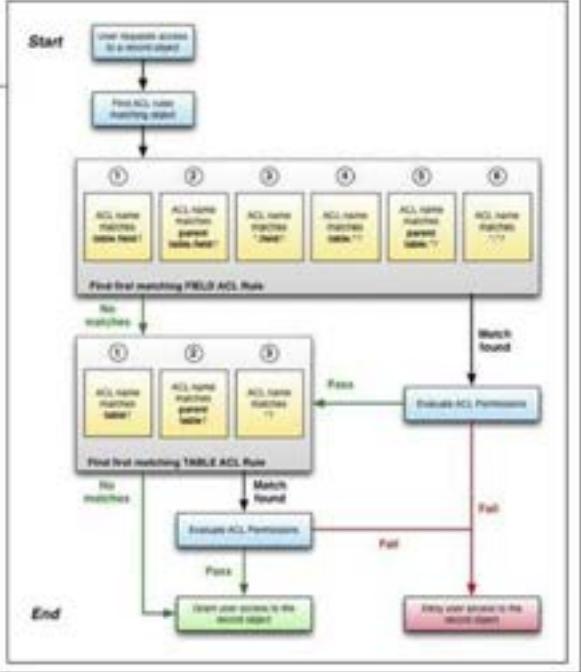
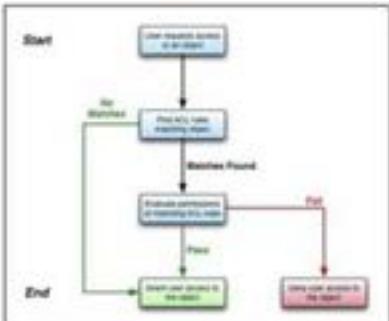
#	Number	Short description	Type	Date	Planned start date
1	1000000000	ITSM update	Normal	None	2017-01-01 00:00:00
2	1000000001	CMDB Asset-ITSM Integration Update Asset	Normal	None	2017-01-01 00:00:00
3	1000000002	ITIL	Normal	None	2017-01-01 00:00:00
4	1000000003	Oracle Full Oracle Audit Configuration	Normal	None	2017-01-01 00:00:00
5	1000000004	Java App Remote	Normal	None	2017-01-01 00:00:00

One of the real benefits of using this type of rule is to reduce the amount of rules required to control access, which also results in less required maintenance.

For example, taking the same series of rules above – it could accomplish the same end results without a wildcard rule. However, in order to do so, it would require six separate rules. That is a significant more amount of rules to manage, should changes be needed in the future.

## Access Control Evaluation

One of the most powerful items behind Contextual Security is that it can apply either to a particular field or an entire row, using this process:



Record ACL rules are processed in order of most specific to most general: Match the object against **field ACL rules**. Match the object against **table ACL rules**.

This processing order ensures that users gain access to more specific objects before gaining access to less specific ones. A user must pass both field and table ACL rules in order to access a record object. If a user fails a field ACL rule but passes a table ACL rule, the user is denied access to the field described by the field ACL rule. If a user fails a table ACL rule, the user is denied access to all fields in the table even if the user previously passed a field ACL rule.

In most cases there is not an individual field ACL rule for every field in the table the user is trying to access. If no field ACL rule matches the record object, the user must pass the table ACL rule. Since the base platform includes wildcard table ACL rules that match every table, the user must always pass at least one table ACL rule. The base platform provides additional table ACL rules to control access to specific tables.

Table ACL rules are processed in the following order: Match the table name. For example, incident. Match the parent table name. For example, task. Match any table name (wildcard). For example, \*. Just like with field ACL rules, the platform grants the user access to the record object secured by the ACL rule and stops searching for matching ACL rules the first time a user passes a table ACL rule's permissions. A user who passes the table ACL rule for Incident has access to all fields in the Incident table. A user who passes the table ACL rule for task has access to all fields in the Task table as well as the fields in extended tables. A user who passes the table ACL rule for any table has access to all fields in all tables.

### Section Summary

- User Permissions Summary
- What is an Access Control?
- Access Control List (ACL)
- Rule Types
- Rule Definition Criteria
- Evaluation Workflows



#### Lab 3.5 – Data Security:

- Provide application menu and module access for a specified role
- Create an Access Control rule to grant permission

**Lab Dependency** – This lab requires the completion of Lab 3.1.

## Lab Topics

- Provide application menu and module access for a specified role
- Create an Access Control rule to grant permission

**Lab Dependency** – This lab requires the completion of Lab 3.1.

After much deliberation, Cloud Dimensions management have decided to limit access to Infinity device data to only those teams actively supporting the product.

The result is a requirement to restrict access to the Infinity Inventory application menu and modules.

As the system administrator, you will act upon the requirements provided by Cloud Dimensions management to successfully secure Infinity device data.

## Lab 3.5 Data Security

### Update Table Role

1. System Security > Users and Groups > Roles.
2. Search for and open the **u\_infinity\_user** role record.

**NOTE:** This role was automatically created when the **Infinity** table was added to the database, along with the creation of four table Access Control rules. All of these are optional during the table creation process and can be avoided through settings.

3. Change the name to **u\_infinity\_support**.
4. Type into the **Description** field: **Support role for the Infinity**.
5. From the **Form Control Menu**, select **Insert**.

**NOTE:** Two infinity roles now exist which will allow for more granular access provisioning in the future:

Name
u_infinity_support
u_infinity_user

## Confirm Access to the Infinity Inventory Application

1. System Definition > Application Menus.
2. Use any method to find and open the **Infinity Inventory** application menu.
3. Notice the value next to the Roles field, **u\_infinity\_user**:

A screenshot of a software interface showing an application menu record for 'Infinity Inventory'. The 'Title' field contains 'Infinity Inventory'. The 'Roles' field contains 'u\_infinity\_user', which is highlighted with a red box. To the left of the roles field is an edit icon (pencil).

4. Click the **Edit User Roles** icon (pencil) next to the Roles field:

A screenshot of the same software interface as above, but now the edit icon (pencil) next to the 'Roles' field is highlighted with a red box.

5. Add the **u\_infinity\_support** role from Available to Selected.
6. Click the **Done** button.
7. **Update** the Application Menu record.

## Update the Infinity Support Group

1. User Administration > Groups.
2. Locate and open the **Infinity Support** group record.
3. Select the **Roles** tab, if needed, then click **Edit...**.
4. Add the **u\_infinity\_support** role to the Roles List.
5. **Save**.

## Test the Visibility Settings

1. Impersonate Rita Center.

2. Confirm **Rita Center** has been denied access to the **Infinity Inventory** application and to the modules it contains.
3. Impersonate **Kevin Edd**.
4. Expand the **Infinity Inventory** application menu and notice **3** modules display:



**NOTE:** A module is missing from the application menu. Which one?

5. Click on the **Employees** module.

Although **Kevin Edd** has inherited the **u\_infinity\_support** role because he is a member of the **Infinity Support** group, which has access to the **Infinity Inventory Application**, the role does not currently have access rights to the underlying data.

As a result, the message "Security constraints prevent access to requested page" appears when Kevin attempts to access the Infinity table's data.

Access Control rules can be created to allow users with a certain role access to the table records, but first let us update Rita and Kevin's group permissions so Infinity Support and Infinity Customer Support group members are granted access to the application and all of its modules.

## Update Roles and Groups

1. As System Administrator, **User Administration > Roles**.
2. Locate and open the **u\_infinity\_support** record.
3. Under the **Contains Roles** section, click **Edit...**
4. Add **u\_infinity\_user** to the **Contains Roles List**.
5. Click **Save**.

**NOTE:** You've added the **u\_infinity\_user** role and its permissions under the **u\_infinity\_support** role. All users with the **u\_infinity\_support** role now automatically inherit the permissions found with **u\_infinity\_user**.

6. **User Administration > Groups.**
7. **Open Infinity Customer Support.**
8. Under Roles, click **Edit...**
9. Add the **u\_infinity\_user** role.
10. **Save.**
11. **Impersonate Rita Center.**
12. Confirm the **Infinity Inventory** application menu is accessible and its four modules display.

**NOTE:** You will also now see data when you select any of the modules.

### Create an Access Control Rule

Now create an Access Control rule that allows only the System Administrator to update the **Asset tag** field on the Infinity record, and no other role.

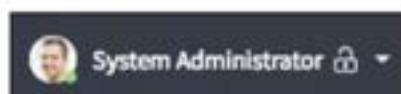
1. Impersonate the **System Administrator**.
2. Open the **User** menu.
3. Next, choose **Elevate Roles**:



4. Select the checkbox next to **security\_admin**:

<input checked="" type="checkbox"/>	security_admin
-------------------------------------	----------------

5. Click **OK**.
6. Notice the page refreshes and there is now an unlocked pad icon next to your name on the User menu:



7. **System Security > Access Control (ACL)**.
8. Filter the list of rules to find the four **u\_cmdb\_ci\_infinity** rules.
9. Open the rule with the **write** operation:

Name	Operation
u_cmdb_ci_infinity	write

10. From the **Name** field, select the drop down next to the drop down with the value **Infinity [u\_cmdb\_ci\_infinity]**:

A screenshot of a dropdown menu. It shows a list item "infinity [u\_cmdb\_ci\_infinity]" with a red box around it, indicating it is selected or highlighted.

11. From the drop down, select **Asset tag**.
13. Open the **Form Control Menu** and select **Insert and Stay**.
14. From the **Verify Security Rules** window, click **Continue**.
12. Scroll down to the **Requires role** section.
13. Double click on **Insert a new row...**.
14. Type **admin** and click the save icon to add the role.
15. **Update**.

**NOTE:** The **u\_cmdb\_ci\_infinity.asset\_tag** rule was created and added to the Access Control List.

## LAB VERIFICATION

### Test New Security Settings

- As Rita Center, Infinity Inventory > Add Inventory:

The screenshot shows the 'Add Inventory' form in ServiceNow. It includes fields for Device Number, Name, Asset tag, Device version, Support group, Installed, Status, Owned by, Email, Company, and Device Notes. The Asset tag field is grayed out, indicating it is not editable. The Submit button is visible at the top right.

**NOTE:** All fields but Asset tag, Email, and Company should be editable and you should see the **Submit** button.

- Infinity Inventory > All Devices:

	Device Number	Name	Owned by	Location	Status	Support group	Updated
	C000000001		Rita Center	1234 High Point Road, Unithomas, NC	Installed	infinity Support	2017-07-20 12:37:24
	C000000002	infinity Alpha Prototype	Sierra Beach	1234 Cloud Avenue, Del Lago, CA	Installed	infinity Support	2017-07-20 12:36:03
	C000000003	infinity Alpha Prototype	Sierra Beach	1234 Sunlit Sierra Avenue, Sierra Beach, CA	Installed	infinity Support	2017-07-20 12:36:03
	C000000004	infinity Testing Device	Sierra Beach	Rocky Mountain Landstraße 222, Frankfurt	Installed	infinity Support	2017-07-20 12:37:24

**NOTE:** You should see 19 records.

3. Opening any Infinity Inventory record:

The screenshot shows a ServiceNow form titled "Infinity - Infinity Alpha Prototype". The form contains the following fields and their values:

Field	Value
Device Number	CD100000
Name	Infinity Alpha Prototype
Asset tag	(disabled)
Device location	(disabled)
Support group	Infinity Support
Created	(disabled)
Status	checked
Owned by	Agent-SAM
Email	Agent-admin@example.com
Company	Cloud Dimensions
Device Notes	(empty)

At the bottom of the form, there is a "Related items" section with a search bar and some navigation icons.

**NOTE:** All fields but Asset tag, Email, and Company should be editable.

4. Impersonate System Administrator.

*Congratulations, you have completed the Data Security Lab!*

## Module Recap: Data Administration

servicenow

**Why** would you use these capabilities?

**When** would you use these capabilities?

**How often** would you use these capabilities?

Data Schema	CMDB	Import Sets	Reporting	Data Security
Table Extensions	CMDB	Import Sets	Reporting	Table.None
Tables & Columns	Configuration Items	Transform Maps	Report Distribution	Table.field
Schema Map	Dependency Views	Coalesce Fields	Metrics	Table.*
Data Archiving			Dashboards	

service**now**

- 1
- 2
- 3
- 4**
- 5

## Module 4

### Service Automation

- 4.1 Knowledge Management
  - 4.2 Service Catalog
  - 4.3 Workflows
  - 4.4 Service Level Agreements (SLAs)
-

## Module 4

service*now*

### 4.1 Knowledge Management

- What is Knowledge Management?
- Knowledge Base Architecture
- Knowledge Workflows
- Article Feedback
- Knowledge Security

## What is Knowledge Management?

servicenow

**Knowledge Management** allows users to create, edit, and view Knowledge articles to share information across the organization in a centralized location

Administrators and those with the **knowledge\_admin** role can create multiple Knowledge Bases

Knowledge articles exist within a single Knowledge Base, which is managed by one or more Knowledge Managers



With Knowledge Management, each organization can have their own Knowledge Base with flexible controls over who can see the information and who can help contribute.

Users can browse and search Knowledge using a Knowledge Homepage or use contextual search from other ServiceNow applications, so users can help themselves troubleshoot their issue.

To view Knowledge content, navigate to **Self-Service > Knowledge** to displays knowledge articles organized by Knowledge Base and Category, as well as Featured Content, and popular articles (Most Useful and Most Viewed).

From the Knowledge homepage you can browse and search for articles, then submit feedback on those articles if desired.

**NOTE:** You must have at least one ServiceNow role to contribute content.

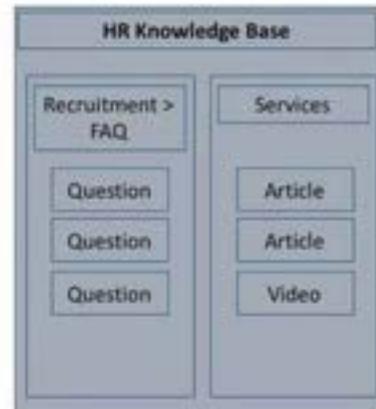
## Knowledge Base Architecture

servicenow

### Category Hierarchy



### Knowledge Base Examples



Administrators can create multiple Knowledge Bases and assign them to individual managers responsible for controlling the behavior and organizational scheme of each knowledge base. Each Knowledge Bases can have unique lifecycle workflows, user criteria, category structures, and management assignments.

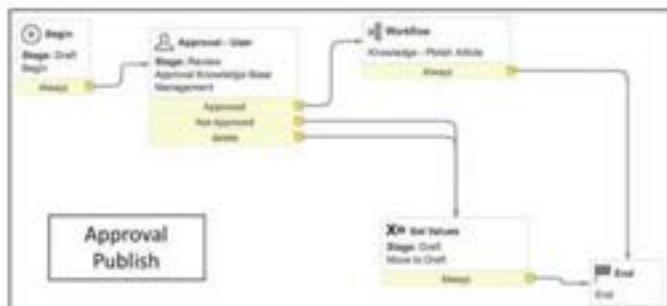
#### Category Hierarchy:

- Knowledge articles within a Knowledge Base are grouped by category
- Category groups can help you define the Knowledge Base taxonomy, and can help users find articles within a Knowledge Base
- Knowledge Managers can define knowledge categories to pre-populate the list of available categories, and knowledge contributors can select categories, and add or edit categories, if enabled, for a Knowledge Base

## Knowledge Workflows

servicenow

The publishing and retirement processes for a Knowledge article are controlled by Workflows defined for the Knowledge Base that the article belongs to



You can assign different Workflows to each Knowledge Base

You can use one of the default Workflows, or create your own Workflow to define custom publishing and retirement processes for different types of knowledge

The Knowledge Workflows available in the ServiceNow base instance include:

- **Knowledge – Approval Publish:** Requests approval from a manager of the Knowledge Base before moving the article to the published state. The Workflow is canceled and the article remains in the draft state if any manager rejects the request.
- **Knowledge – Instant Publish:** Immediately publishes a draft article without requiring an approval
- **Knowledge – Instant Retire:** Immediately retires a published article without requiring an approval
- **Knowledge – Retire Knowledge:** Moves a knowledge article to the retired state

**NOTE:** This is only a selection of the base instance Workflows to choose from, as designed for Knowledge Base management.

## Article Feedback: Flag an Article

servicenow

You can search for Knowledge articles from the Knowledge homepage using the search bar at the top of the page

The screenshot shows a search interface with a search bar containing 'email'. Below the search bar is a 'Filter by' section with dropdown menus for 'Type', 'Knowledge Base', 'Categories', and 'Authors'. The 'Type' dropdown is set to 'Knowledge Article'. The 'Knowledge Base' dropdown is set to 'IT'. The 'Categories' dropdown includes 'Email', 'Email > Outlook > Outlook 2010', 'IT > Announcements', 'Operating Systems > Windows', and 'Authors' with options like 'Mike Carter' and 'Mike Hartling'. The 'Authors' dropdown is currently selected. The main area displays 'email Search Results' with four article cards. The first card is titled 'Delete Email Recovery' and discusses how to delete emails from your CRM. The second card is titled 'What is Spammer?' and explains what spam is and how to identify it. The third card is titled 'How to Block Email Spammers' and provides instructions on blocking spam. The fourth card is titled 'What are phishing scams and how can I avoid them?' and discusses the risks of phishing.

Articles can be flagged for review by using the **Flag article** button to enter feedback

This screenshot shows a single knowledge article record. At the top right, there is a red box around the 'Flag article' button. The article title is 'Managing Settings in Internet Explorer 10 for Windows 8'. It has a 'Number' field showing 'KBR00000023', a 'Last updated' field showing '2 hrs ago', and a 'Rating' field showing '0'. Below the article title, there is some descriptive text about managing settings in Internet Explorer 10 for Windows 8.

To review flagged articles navigate to **Knowledge > Articles > All Flagged**

In the example shown here, a search for email is displayed. Search results include only articles the current user can read and you can select only Knowledge Bases you can access.

Search results can be sorted by relevancy, last updated, or number of views. Results can be filtered in various ways:

- **Knowledge Bases:** Select a Knowledge Base to search. If you do not select a specific Knowledge Base, the search results include articles from all Knowledge Bases you can access. You can also select a Knowledge Base from the choice list in the search bar.
- **Categories:** You can select one or more knowledge categories
- **Authors:** You can select one or more authors to view the articles they have written

Knowledge content is segmented in Knowledge Bases with categories and infinite subcategories.

The Flag Article button displays when a published article is viewed. To view feedback left on a published article, navigate to **Knowledge > Articles > My Flagged** or the **All Flagged** module and open the article record. Alternatively, from the Knowledge list (**Knowledge > Articles > Published**), click the article link in the **Number** column and scroll down to the **Knowledge Feedback** section to view a comment.

Flagging an article allows a user to submit feedback about the article to Knowledge Managers. It is meant to provide feedback in an escalated manner, as opposed to adding a comment to the article, to help improve overall quality of knowledge content.

## Knowledge Security: User Criteria

servicenow

**User Criteria** defines conditions that are evaluated against users to determine which users can create, read, write, and retire knowledge articles

You can apply several user criteria records to knowledge content

User Criteria is applied at the Knowledge Base level

The screenshot shows a ServiceNow interface for creating a new User Criteria record. The title bar says 'User Criteria' and 'The ACME North America HR Department'. The form has fields for 'Name' (set to 'The ACME North America HR Department'), 'Application' (set to 'Global'), and 'Match All' (checkbox checked). Below these are sections for 'User' (checkbox checked), 'Group' (checkbox checked), 'Role' (checkbox checked), and 'Advanced' (checkbox unchecked). To the right, there are dropdown menus for 'Company' (set to 'ACME North America'), 'Location' (dropdown menu), 'Department' (dropdown menu), and 'Match All' (checkbox checked).

Knowledge Bases use User Criteria records to determine which sets of users can read or contribute knowledge within that Knowledge Base. If a knowledge base has no user criteria selected, articles within that Knowledge Base are available to all users.

User Criteria definitions include:

- **canRead**: users who can read all Knowledge Base articles
- **cantRead**: users who cannot read, create, or modify articles in the Knowledge Base
- **canContribute**: users who can read, create, and modify articles in the Knowledge Base
- **cantContribute**: users who cannot create or modify articles in the Knowledge Base

To implement User Criteria, navigate to **Knowledge > Knowledge Bases** and select a Knowledge Base. Next, access the **Can read** or **Can contribute** related lists to select or create User Criteria records.

When creating User Criteria, the **Match All** check box to determine whether all elements from each populated criteria field must match. If selected, only users who match all criteria are given access. If cleared, the user must meet one or more of the set criteria to be given access. By default, this check box is cleared so that any condition met provides a match.

### Section Summary

- What is Knowledge Management?
- Knowledge Base Architecture
- Knowledge Workflows
- Article Feedback
- Knowledge Security



#### Lab 4.1 – Knowledge Management:

- Create a Knowledge Base article by importing a Word document
- Approve the article for publishing
- Define, apply, and test user criteria on the knowledge base

# Lab Topics

- Create a Knowledge Base article by importing a Word document
- Approve the article for publishing
- Define, apply, and test user criteria on the knowledge base

After a few positive rounds of Infinity testing, the product has been greatly improved and the testing audience is ready to be expanded.

Human Resources (HR) has volunteered to lead an initiative within Cloud Dimensions to advertise Open Enrollment for Infinity testing using the Knowledge Base.

HR representatives will be granted the appropriate authoring permissions by the system administrator so that they may create, review, and publish articles.

The System Administrator will also assist with ensuring the article remains secure and accessible only by employees of Cloud Dimensions.

## Lab 4.1 Knowledge Management

### Download Prototype Security Requirements Word Document

You need to download a MS Word document, **Infinity\_Open\_Enrollment.docx**, and save it to your desktop. Your instructor will tell you where you can find the file.

### Create a New IT Knowledge Base Article

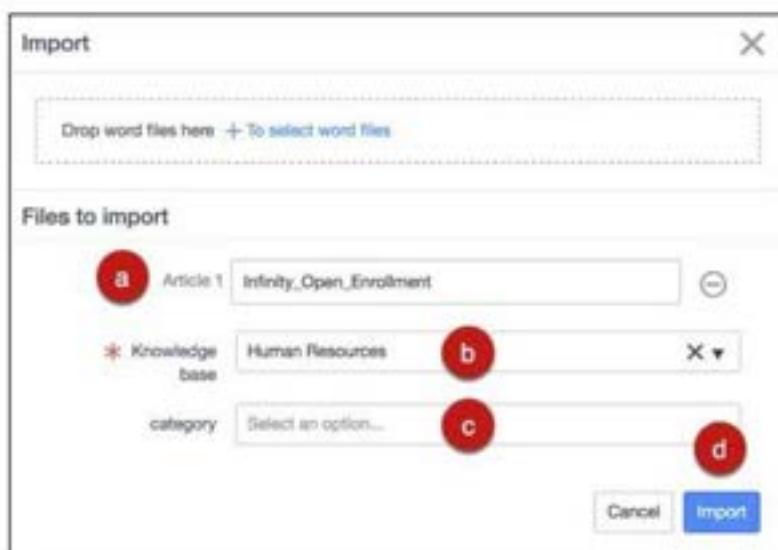
1. Impersonate Jasmin Gum.

**NOTE:** Jasmin Gum is a member of the Human Resources group.

2. Self-Service > Knowledge.
3. Click the **Import Articles** button:



4. Next, click + To select word files and find the **Infinity\_Open\_Enrollment.docx** document you have downloaded, then fill out the form as shown:
- Article 1: **Infinity\_Open\_Enrollment** (auto fills)
  - Knowledge base: **Human Resources**
  - category: [leave empty]
  - Click the **Import** button



**NOTE:** You should receive an “Import completed” message on the bottom of your screen.

5. **Self-Service > My Knowledge Articles.**
6. Click the KB article **Number** for **Infinity\_Open\_Enrollment**:

Knowledge		New	Go to	Number	Search	✓	<	<<	<<<	1 to
		All	> Workflow in (Draft, Review, Published, Pending retirement, Retired)	> Author = Jasmin Gumm						
		Number	Short description	Author	Category	Workflow				
		KB0010001	Infinity_Open_Enrollment	Jasmin Gumm						

7. Update the **Short description** to **Infinity Testing Open Enrollment**.
8. Click the **Publish** button from the header.

**NOTE:** The default publish workflow for the Human Resources Knowledge Base is **Knowledge – Approval Publish**. This means after an author clicks the **Publish** button

on their article, it goes into a review state. Other users with the correct permissions can view the article and determine if any changes are needed before approving and publishing the article.

## Approve the Article for Publishing

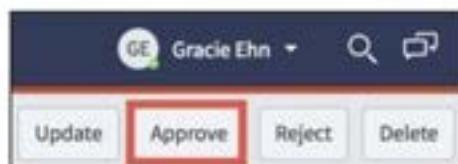
1. Impersonate **Gracie Ehn**, manager of Jasmin Gum.

**NOTE:** Gracie Ehn is also a member of Human Resources and acts as the review board for new content submitted to be published.

2. **Service Desk > My Approvals.**
3. Locate and open the requested approval record for the Infinity Testing Open Enrollment knowledge article:

The screenshot shows a list of approvals. One item is selected, showing details: "All > Approver = Gracie Ehn". The row contains columns for State (Requested), Approver (Gracie.Ehn), and other status indicators.

4. Scroll down to see a summary of the item being approved.
5. Assume the content looks good, then click **Approve** from the form header:



6. **Self-Service > Knowledge.**
7. Open the Human Resources Knowledge Base to confirm the article appears:

The screenshot shows the Knowledge Base search results for "Human Resources". A specific article titled "Infinity Testing Open Enrollment" is displayed, showing its author (Jasmin Gum), views (3), last update (3m ago), rating (4 stars), and a snippet of its content: "Infinity Testing Open Enrollment What is Infinity? Infinity is Cloud Dimensions latest product offering, a portable holographic projector (PHP for short), which is capable of projecting immersive environments around you such as a rain forest, grandma's...".

## Create and Apply User Criteria

The Human Resources Knowledge Base is currently public to all users that log into the Cloud Dimensions instance and access the **Self-Service > Knowledge** module. The system administrator will create user criteria and apply it to the Knowledge Base to appropriately control who can view the content.

1. Impersonate **System Administrator**.
2. **Knowledge > Administration > User Criteria**.
3. **New**.
4. Fill out the form as follows:

Name: **Cloud Dimensions Employees**  
Companies: **Cloud Dimensions**

5. **Submit**.
6. **Knowledge > Administration > Knowledge Bases**.
7. Next, open the **Human Resources** record.
8. Scroll down and click the **Can Read** tab:



9. Click **Edit...**.
10. Add **Cloud Dimensions Employees** to the Can Read List.
11. **Save**.

## Test User Criteria

1. Impersonate **Jon Floyd**.

**NOTE:** Remember, Jon Floyd works for a partner company of Cloud Dimensions. We will use his user account to verify the Human Resources Knowledge Base is unavailable to access because of the user criteria applied to it.

## 2. Self-Service > Knowledge.

The Human Resources Knowledge Base does not appear.

## LAB VERIFICATION

### Human Resource Knowledge Base Article

The screenshot shows a ServiceNow knowledge base article. At the top, there's a header bar with icons for Home, Search, Flag article, Create incident, and a blue 'Save' button. Below the header, the title 'Infinity Testing Open Enrollment' is displayed, along with a reference number 'K00010002' and a status '1 views'. The main content area contains several sections: 'What is Infinity?' which describes it as a portable holographic projector (PHP) capable of projecting immersible environments around you such as a rain forest, grandma's house, and even a lunar landscape; 'What does this open enrollment mean?' which states that Infinity has been tested many times over by teams that are closest to it - multiple Cloud Dimensions teams that support Infinity have had their hands on it for a few months now; and a note at the bottom stating 'Because of how well the Infinity devices are shaping up, we want to extend the testing audience to all Cloud Dimensions employees.'

### User Criteria – Employee Perspective

This screenshot shows the 'Knowledge Bases' section from the Employee Perspective. It displays a category for 'Human Resources' with a sub-section for 'Knowledge'. Under 'Knowledge', there is a box containing '0 Questions and 1 Articles'. The '1 Articles' link is highlighted with a red box.

### User Criteria – Non-Employee Perspective

This screenshot shows the 'Knowledge Bases' section from the Non-Employee Perspective. It displays a category for 'Knowledge' with a sub-section for 'Knowledge'. Under 'Knowledge', there is a box containing '0 Articles'.

**Congratulations, you have completed the Knowledge Management Lab!**

## Module 4

service*now*

### 4.2 Service Catalog

- What is the Service Catalog?
- Major Components
  - Items
  - Record Producers
  - Variables/Variable Sets
  - Workflows
- Order Guides
- Service Catalog Security

## What is the Service Catalog?

service<sup>now</sup>

The **Service Catalog** is a robust ordering system for services and products offered by various departments for users:

- An ordering system for goods and services
- Categories organize catalog items
- “One stop shopping” for users
- Access to the Service Desk
- Help and Training Portal

Multiple Catalogs are supported



The IT Service Catalog lets users see a list of things they may need (to create a request for), or would like to have from IT. Administrators and users with the catalog admin role can define catalog items, including formatted descriptions, photos, and prices.

Categories define the organization for Service Catalog items. Categories organize service catalog items into logical groups. Categories can have a parent-child relationship, such as **IT** and **Laptops**. A child category is a subcategory of its parent category. Each Catalog Item, Order Guide, Record Producer, Content Item, and subcategory appears as a single item within the category.

In Service Catalog, Employee Self Service (ESS) users can order pre-defined, bundled goods and services from the IT organization, or other departments. ITIL users or Administrators see additional choices; Administrators can view all the Service and Item categories. **Top Requests** is a dynamic category displaying the five most ordered items.

The ServiceNow platform supports multiple Service Catalogs. System Administrators and Catalog Administrators can manage multiple Service Catalogs and provide services to different teams within the organization. Examples include: IT Services, Human Resources, and Facilities Management. Views can be defined for groups that view a Catalog, and Catalog Items can be shared by multiple Catalogs.

## Service Catalog Major Components

service<sup>now</sup>

Items	Record Producers	Variables	Variable Sets	Workflows
				

Items are the building blocks of the Service Catalog

Record Producers are a form that produce a task record

Variables provide questions to help the requestor specify what item, option, or service to order

Variable Sets are a modular unit of variables that can be shared between catalog items

Workflows run behind the scenes and communicate the stages of the approval process to the requestor, as well as drive the request fulfillment

**Items:** In the Service Catalog, users locate a category for an item or service they want to order, and then click the subcategory link.

**Record Producer:** An interface used as an alternative to lists and forms. Each Record Producer focuses on a specific process or task and can be used anywhere in the ServiceNow platform. In the Service Catalog, Record Producers are presented in categories along with catalog items. Users can use Record Producers to create an incident, request an emergency change, and more. This enables the Service Catalog to be used as a complete front-end UI.

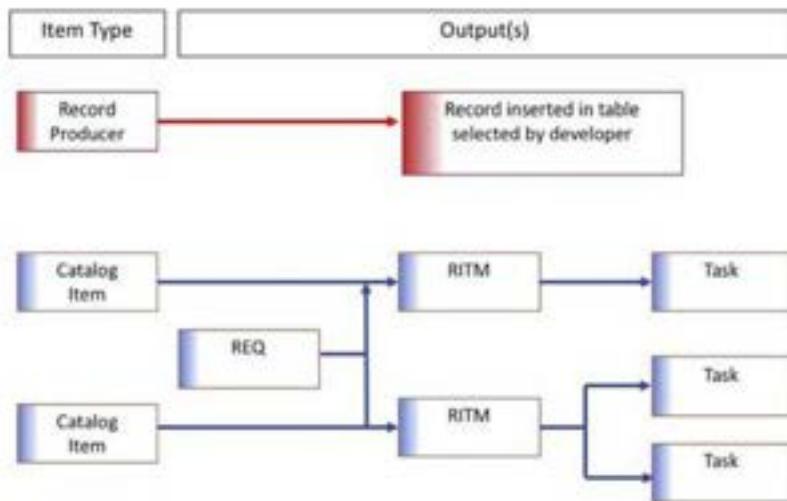
**Variables:** Provide options gather specific information related to the customer's needs. Questions that define item options can be added to ask the end user ordering the catalog item. Variables can affect the order price. Service Catalog variables are flagged as "Global" by default and will display in all the execution tasks of a requested item. A variable is defined once and can be used in multiple places. The Service Catalog allows you to attach variables to a catalog item.

**Variable Sets:** You attach variables either to a Catalog Item or to an Execution Plan.

**Workflows:** When you create a new service catalog item, you can create a new corresponding Workflow at the same time.

## Service Catalog Output

service<sup>now</sup>



This slide shows the difference between a Record Producer and a normal Catalog Item.

Record Producers Service Catalog Items result in records being added or modified in the database, while Service Catalog Items result in Request Items.

## Order Generate Various Numbers

servicenow

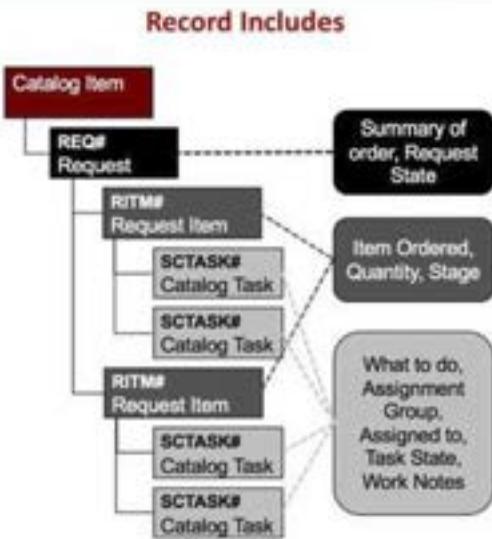
### Example

Service Catalog Option for Ordering Computers

Manager orders 2 new computers for the team

PC 1 – Standard Desktop  
Order Equipment  
Configure Equipment

PC 2 – Notebook  
Order Equipment  
Configure Equipment



A Record Producer focuses on a specific process or task and can be used anywhere in the ServiceNow platform. In the Service Catalog, Record Producers are presented in categories along with catalog items where each table has its own record-identifying designation.

For Catalog Items, a request, an item, and a task are all created when an order is placed, each on a corresponding table:

#### REQ# Request [sc\_request] table:

A request number generated to keep track of an order.

#### RITM# Requested Item [sc\_req\_item] table:

Within a request generated from a catalog order, each discrete item ordered is given a specific "Requested Item Number" known as an RITM (number).

#### SCTASK# Catalog Task [sc\_task] table:

In the Catalog Tasks section for an ordered item, the different tasks display for what has to be done to get the item ready for delivery to the user, for example; the Assignment group, the Due date, Work start, and Work end dates.

## Progress Stages for a Requested Item

service<sup>now</sup>

While viewing the requested item, you can expand the Workflow stages which provide summary-level feedback about the progress or state of an item in the delivery process

Description (Includes Annual Charges)	Delivery Date	Stage	Price (ex.)	Quantity	Total
MacBook Pro	2017-07-18	+  Waiting for Approval (Waiting for Approval (in progress))	\$1,499.00	1	\$1,499.00
		<input type="radio"/> Dept. Head Approval - 2 Days (Dept. Head Approval - 2 Days (Pending - has not started))			
		<input type="radio"/> CIO Approval - 2 Days (CIO Approval - 2 Days (Pending - has not started))			
		<input type="radio"/> Order Fulfillment - 4 Days (Order Fulfillment - 4 Days (Pending - has not started))			
		<input type="radio"/> Deployment - 1 Day (Deployment - 1 Day (Pending - has not started))			
		<input type="radio"/> Completed (Completed (Pending - has not started))			

Some of the progress or state of an item in the delivery process:

- **Waiting for approval (In Progress)**
- **Approved**
- **Pending (has not started)**
- **Fulfillment (In Progress)**
- **Completed**

## Maintain Service Catalog Items and Variables

servicenow

### Maintain Items

Once an item is built (complete with variables and a delivery Workflow), you can do a number of things to maintain that item:

- Publish the item to the Service Catalog, allowing users to order that item
- Group with other items into an Order Guide

### Add Variables

- Add questions/choices to Service Catalog items using Variables
- Variables define the questions to ask the end user ordering the catalog item
- Question choices can define the available options
- Variable options may affect the order price

Service Catalog variables are global by default and provide options to tailor a catalog item to the customer's needs. For example, a computer may be available with different operating systems.

Variables define the questions to ask the end user ordering the catalog item, and Question Choices define the available options. Variable options may also affect the order price.

Use an Order Guide to assist users in determining what items they need.

## About Service Catalog Variables and Display Sequence

servicenow

Variable Types	Variable Sets	Display Sequence (Order)
<p>Common Variable Types</p> <ul style="list-style-type: none"><li>– Multiple Choice</li><li>– Select Box</li><li>– Single Line Text</li><li>– Reference</li><li>– Checkbox</li></ul>	<p>A Variable Set is a modular unit of variables that can be shared between catalog items</p>	<p>The Order field establishes the sequence for displaying lists</p> <ul style="list-style-type: none"><li>– An item with 100 in the Order field, displays first in the list</li><li>– An item with 200 in the Order field, displays second</li></ul>

### Common Variable Types

- **Multiple Choice:** Creates radio buttons for user-defined question choices.
- **Select Box:** Creates a choice list of user-defined question choices.
- **Single Line Text:** Creates a single-line text input field.
- **Reference:** Specifies a record in another table, similar to a reference field.
- **Checkbox:** Creates a checkbox that may be selected or cleared; list checkboxes in order under a label to create an options question.

Functionally, a **Variable Set** is just a container, so it has only two fields: **Name** and **Description**.

The Service Catalog lets you attach variables either to a catalog item or to an execution plan. From the left navigation pane, select **Service Catalog > Catalog Variables > Variable Sets**, and create a new Variable Set.

After you save the Variable Set, you will get a Related List at the bottom where you can add as many variables as you want.

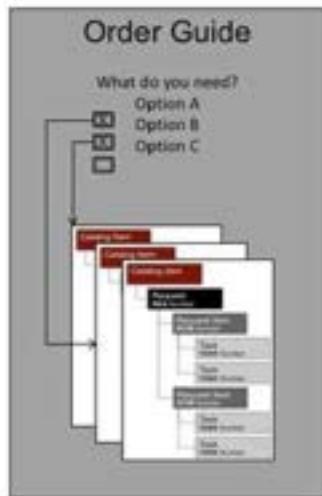
## Order Guide

service<sup>now</sup>

### Best Practices

Define an **Order Guide** to assist customers in ordering a complete set of needed items and to help users see item relationships

Questions can be used to present item options; present users with **only relevant questions and choices** at the appropriate time in the ordering process



Items are the building blocks of the Service Catalog. Once you have built a complete item with variables and a delivery Workflow, you can do a number of things with it, such as add it to an Order Guide.

Order Guides provide the ability to order multiple, related items as one request. Remember that variables are presented by the **Order** field number.

## Service Catalog Security: User Criteria

servicenow

Similar to the knowledge base, user criteria can control access to Service Catalog records

You can apply several user criteria records to a single catalog item or category

The screenshot shows a software interface for creating a new User Criteria record. The title bar says "User Criteria" and "Create with Default Data". The main area has sections for "Name" (set to "Users with Default Role"), "Application" (set to "Global"), and "Active" (set to "Yes"). Below these are dropdown menus for "User", "Group", "Role", and "Advanced". To the right are dropdowns for "Company", "Location", "Department", and "Match All". At the bottom are "Update" and "Delete" buttons.

To apply user criteria to an item or category, open the respective record and navigate to the **Available For** or **Not Available For** related lists.

**NOTE:** These related lists are not on the form by default and must be added by configuring the form.

Next, click **Edit** to add an existing user criteria record, or click **New** to create a new one.

Save the record to associate the user criteria record with the item or category.

**NOTE:** The **Not Available For** settings override the **Available For** settings. A user on the **Not Available For** list cannot access an item or category, even if that user is also on the **Available For** list.

## Section Summary

- What is the Service Catalog?
- Items
- Record Producers
- Variables/Variable Sets
- Workflows
- Order Guides
- Service Catalog Security



### Lab 4.2 – Service Catalog:

- Create an Infinity Service Catalog item
- Add Variables
- Validate the work with "Try It"

# Lab Topics

- Create an Infinity Service Catalog item
- Add Variables
- Validate the work with "Try It"

Up until now, the employee ordering and fulfillment process for an Infinity has been entirely "off the books" – unofficial, to say the very least.

With the availability of the Service Catalog, Cloud Dimensions would like to improve the process and ensure every employee has a chance to receive an Infinity device. Additionally, tracking orders and inventory in one convenient location is appealing.

Infinity is offered to employees in either Crimson or Silver, with optional, additional specifications to choose from.

**NOTE:** You need to download an Infinity image file to your desktop to complete this lab: [Infinity.png](#).

## Create New Service Catalog Item

1. Ensure you are logged into the instance as the **System Administrator**.
2. **Service Catalog > Catalog Definitions > Maintain Items.**

**NOTE:** Although we will complete this lab as the system administrator, any user with the **catalog\_admin** role will be granted access to the **Maintain Items** module and be able to create and edit catalog items.

3. Click **New**.
4. Fill out the top of the form as shown:

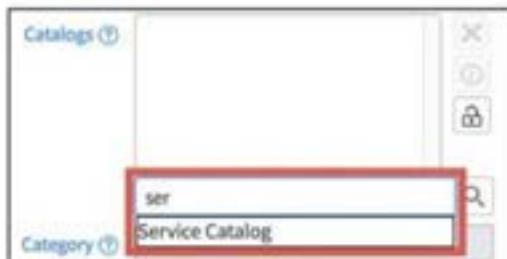
Name: **Infinity**  
Price: **\$395.99 USD**

## Lab 4.2 Service Catalog

5. Click the lock icon (Edit Catalogs) next to Catalogs:



6. Start typing ser then select Service Catalog:



7. In the Category field, select Hardware.

**NOTE:** Category is used to determine where a catalog item appears within the Service Catalog. From the left navigation pane, under **Service Catalog > Maintain Categories**, additional categories can be created and categories can be created in hierarchical fashion.

8. Add a Short Description and Description as shown:

Short description: VR, but without the glasses

Description:

The Infinity is a portable holographic projector (PHP) that is capable of projecting immersive environments around you. What are you waiting for? Get Infinity!

9. In the Picture field, select Click to add...



10. Choose File: **infinity.png** (already downloaded at the beginning of this lab).

11. Click **OK**.

12. Click **Update**.

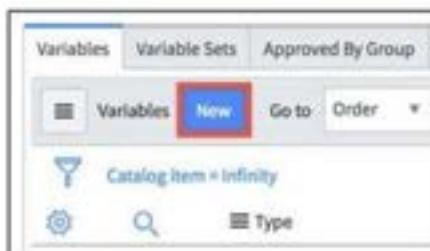
### Verify Service Catalog Item Creation

1. **Self-Service > Service Catalog.**
2. Click the **Hardware** category header.
3. Select the **Infinity** item from the list to open the order screen.

**NOTE:** From this form, a user could order an Infinity, but the form does not yet allow users to specify options for the device.

### Add a Memory Variable to the Service Catalog Item

1. **Service Catalog > Catalog Definitions > Maintain Items.**
2. Locate and open the **Infinity** record.
3. On the **Variables** tab, select **New**:



4. Fill out the form as shown:

Type: **Multiple Choice**  
Mandatory: [select checkbox]

5. Under the **Question\*** tab, enter the Memory question as shown:

Question: **How much memory do you want in your Infinity?**

Name: **memory**

**NOTE:** The corporate-approved version of the Infinity is available in both 256GB and 512GB models.

6. Click the **Default Value\*** tab.
7. Enter the Default value: **256**.
8. Save the form, instead of Submit, to define the Infinity memory Question Choices.
9. Scroll to the **Question Choices** section then click the **New** button.
10. Fill out the **Question Choices** form as shown:

Order: **100**  
Text: **256GB**  
Value: **256**

**11. Submit.**

12. In the **Question Choices** section, click **New** then fill out the form for the second memory choice.

Price: **\$100.00**  
Order: **200**  
Text: **512GB**  
Value: **512**

**NOTE:** The Price field is utilized for the 512GB choice, as it adds \$100 to the overall price. Additionally, the order value is set to 200 which will place the 512GB memory option second in the choice list.

**13. Submit.**

14. You should now see both question choice values:

Question = How much memory do you want in your Infinity?		
	≡ Text	≡ Value
①	256GB	256
①	512GB	512

**NOTE:** This variable definition is now complete. It provides the end user with a choice of memory options for their Infinity catalog request.

15. Click **Update** to return to the Infinity item record.

## Add Color Variables to Service Catalog Item

You will now create a second question for a color choice by configuring a second variable.

1. In the **Variables** related list, click **New**.
2. Complete the form as shown:

Type: **Select Box**

In the **Question\*** tab,  
Question: **What color Infinity would you like?**  
Name: **color**

In the **Default Value** tab,  
Default value: **crimson**

3. Save.
4. Scroll to the **Question Choices** section then click the **New** button.
5. Fill out the form for the first color choice:

Text: **Crimson**  
Value: **crimson**

6. Save.
7. Change the following field values for the second color choice:

Order: **200**  
Text: **Silver**  
Value: **silver**

8. Open the **Form Control Menu**, then select **Insert** to return to the color variable record.

**NOTE:** You have just defined the values for the **What color would you like?** variable, using question choices.

Another option for providing value choices is to use reference tables or fields from the database. See the **Choice table** and **Choice field** options under the **Type Specifications** section on the Variable form.

[docs.servicenow.com](http://docs.servicenow.com) defines all of the possible variable types.

9. Click **Update** to return to the Infinity item record.

## LAB VERIFICATION

1. From the top of the **Infinity Catalog Item** form, click the **Try It** button to view the new variable options added.
2. Change the **Infinity** color to **Silver**.
3. Choose **512 GB** and notice how the item **Price** changes:

The screenshot shows a Service Catalog item page for 'Cloud Dimensions INFINITY'. At the top, there's a navigation bar with 'Service Catalog > Hardware > Infinity'. Below the title, it says 'VR, but without the glasses'. On the left, there's a sidebar with 'What color Infinity would you like?' showing 'Silver' selected. Under 'How much memory do you want in your Infinity?', '512GB' is selected. The main content area describes the Infinity as a portable holographic projector (PHP) capable of projecting immersive environments. It includes a 'Get Infinity!' button. To the right, there's a summary table with 'Order this item' sections for Price (\$499.99), Quantity (1), Subtotal (\$499.99), Delivery time (3 Days), and buttons for 'Order Now' (highlighted in blue), 'Add to Cart', and a 'Shopping Cart' section showing 'Empty'.

*Congratulations, you have completed the Service Catalog Lab!*

## Module 4

servicenow

### 4.3 Workflows

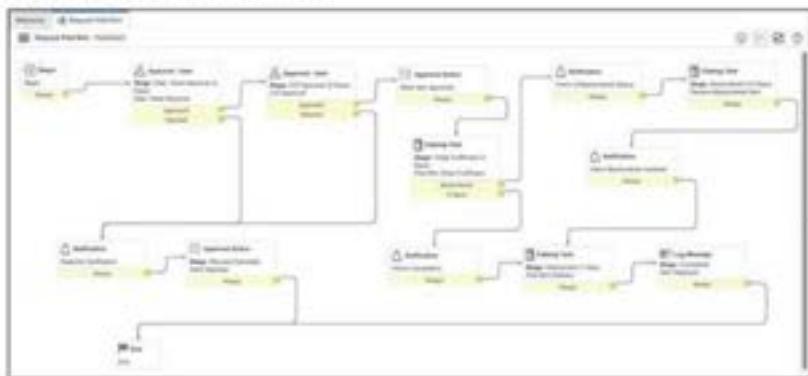
- What is a Workflow?
- Using Workflows
- Starting Workflows
- Terminology
  - Model
  - Version
  - Activities
  - Conditions
  - Stages
- Workflow States
- Workflow Editor

servicenow

## What is a Workflow?

servicenow

A **Workflow** is a virtual representation of activities consisting of connected steps planned out in a sequential manner



This is a Workflow **model** which represents the entire structure of the Workflow and the definition of the processing a Workflow can accomplish

The graphical Workflow Editor provides a drag-and-drop interface for automating multi-step processes across the platform. Each Workflow consists of a sequence of activities, such as generating records, notifying users of pending approvals, or running scripts.

The Workflow starts when a triggering event occurs. Common triggers include a record being inserted into a specific table, or a particular field in a table being set to a specified value. For example, you might create a Workflow that runs whenever a user requests approval for an item they want to order from the catalog. When an activity completes, the Workflow transitions to next activity. An activity might have several different possible transitions to various activities, depending on the outcome of the activity. Continuing the example above, if the user's request is approved, the activity might transition to an activity that notifies someone to order the item; if the user's request is denied, the activity might transition to notifying the user that their request has been denied.

The graphical Workflow Editor represents Workflows visually as a type of flowchart. It shows activities as boxes labeled with information about that activity and transitions from one activity to the next as lines connecting the boxes.

An activity is processed for each step in the Workflow (the "behavior" is determined when the activity begins).

When this action completes, the Workflow checks each of the activity's conditions. For each matching condition, the Workflow follows the transition to the next activity. When the Workflow runs out of activities, the Workflow is complete.

## When to Use a Workflow

service*now*



### Use a Workflow

To automate a repeatable process

When a standard response is needed for every record insert, update, and delete

When automation is needed in the platform (not limited to Service Catalog and SLAs)



### Do Not Use a Workflow

When doing a simple discrete record change

If the process being defined does not have a standardized response (new processes that are being refined)

service*now*

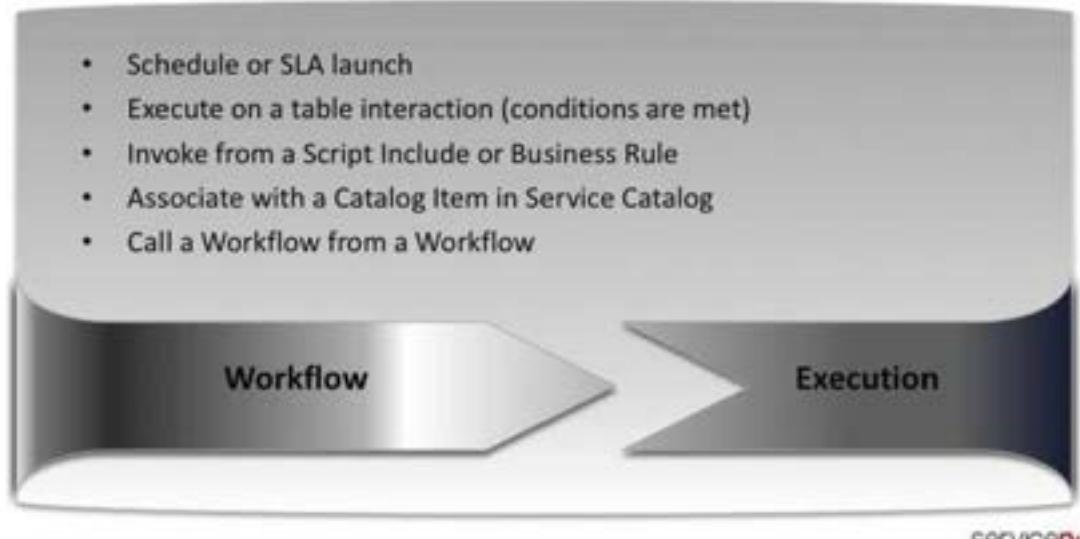
You can create a Workflow with the workflow editor to automate a multi-step process.

**NOTE:** You must have the `workflow_admin` or `workflow_creator` role to use the workflow editor.

## How Can Workflows be Started?

servicenow

- Schedule or SLA launch
- Execute on a table interaction (conditions are met)
- Invoke from a Script Include or Business Rule
- Associate with a Catalog Item in Service Catalog
- Call a Workflow from a Workflow



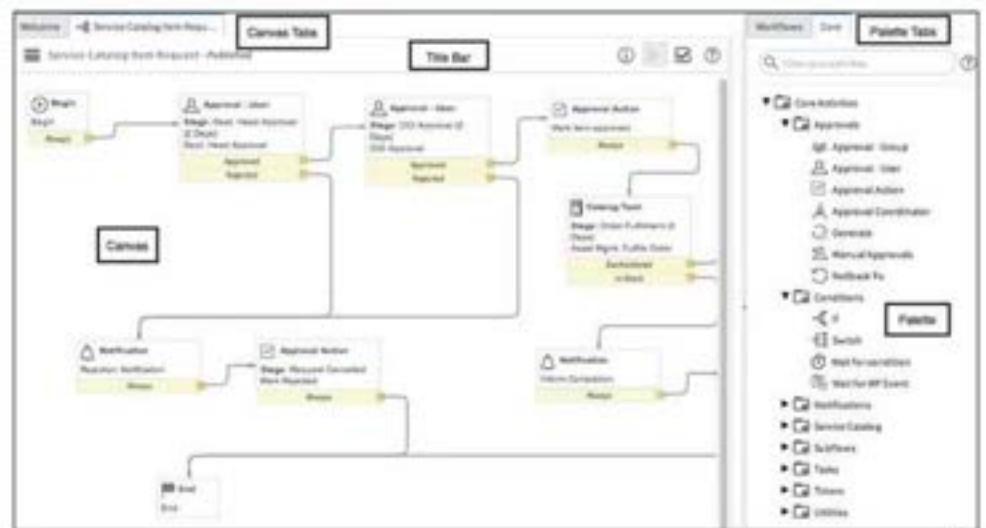
Only published workflows can be instantiated. Unpublished workflows can not be started.

### Notes about Workflow execution:

- If the processing conditions are met, a Workflow can be automatically started on a table record insert or update operation.
- Workflows that are built for the **Requested Item [sc\_req\_item]** table can be manually associated with a Service Catalog Item. Once associated, the Workflow will start automatically. This is also true for SLA Workflows.
- The **startFlow(workflowId,current,operation,vars)** method is one method that can be used to call a Workflow from a Script Include or Business Rule.
- Workflows that are called from other Workflows are called subflows to that Workflow.

## What is the Workflow Editor?

service<sup>now</sup>



service<sup>now</sup>

The Workflow Editor is an interface for creating and modifying workflows by arranging and connecting activities to processes. Activities can be added, removed, or rearranged. Transitions can be drawn or changed, and the workflow powers the process as defined visually. All workflows need to have a **Begin** and an **End**.

The Workflow editor user interface is divided into the following areas:

- **Canvas Tabs:** Contains tabs for accessing workflows being edited or created
- **Title Bar:** Displays the workflow name and status. Provides a menu and controls for configuring, testing, and validating workflows.
- **Canvas:** Provides the working surface for creating new workflows or editing existing ones
- **Palette Tabs:** Contains all available workflow activities and existing workflows you can use as subflows, displayed in the **Palette**. Drag activities and subflows to the canvas to create new workflows or edit existing ones.

When you open an existing workflow from the main ServiceNow interface, the Workflow Editor opens in a new browser tab.

Workflows need to be checked out before they can be edited.

When a Workflow is checked out, changes only apply to the user who has the Workflow checked out. Other users can continue to use the published Workflow. After the changes are complete, the Workflow can be published so that it is available to all users.

## Workflow Terminology

servicenow

Workflows use conditions and a table association to determine when to start, execute, continue, and end

Workflows have automation blocks called **Activities** that are connected by **transitions** (lines) that establish possible processing paths

Workflow **version** is the currently published **model** that is available to the instance

Workflow **context** is the instantiation of the version that is executing for a given record, showing the processing path executed

servicenow

An **Activity** is a Workflow block that organizes the individual actions the Workflow performs as it runs. Almost like a customizable API, Activities are the organizational element that defines the individual actions the Workflow performs as it runs. Plugins add specialized Activities to the Activities list and custom Activities can be created. A transition, or line, establishes the processing paths or order in which Activities are executed.

To design a Workflow model, drag an Activity into the Workflow Editor and connect it to other Activities by drawing transitions. You can also copy an existing Workflow to add, delete, and connect Activities. In the Workflow Editor, scrollbars are used to view parts of the Workflow and the **Expand Transitions** command to redraw lines.

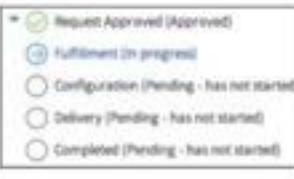
A Workflow version maintains these properties:

- The Workflow table association
- The Conditions, Published Status, and permissions of the Workflow. There can only be one active version at any given time.
- The Workflow Model, which is the Workflow in its entirety as the set of Workflow Activities and their transitions (lines)

Contexts can use different versions of the same Workflow, and all of these contexts can be running at the same time.

## Workflow Activities, Conditions, and Stages

servicenow

Activities	Conditions	Stages
An <b>activity</b> is a Workflow block that organizes the individual actions the Workflow performs as it runs  	Conditions are activities available for a Workflow  Conditions are used to transition - to move from Activity to Activity, you can add Conditions  	Workflows can provide a summary of Workflow progress by updating any field designated as a Stage field  

servicenow

**Activities** have conditions that route processing. The three basic Workflow Activities are Approval, Notification, and Task. Conditions are the start of a transition to the next activity in the Workflow model.

-- Workflow activities include **Approvals** which allow Workflows to generate and manage approvals while driving a record to fulfillment, **Notifications** which allow Workflows to notify users of events that occur during the Workflow, and **Tasks** which create and modify tasks. These activities are only available when the Workflow is defined to run on a table which extends Tasks. It also includes **Timers** which pause the Workflow for a set period of time, and **Utilities** which provide useful controls over the path of the Workflow, and other useful tools. **Catalog Task** is an activity only available for workflows running against the **Request Item [sc\_request\_item]** table.

**Conditions:** You can add a Condition to an Activity, such as Approval – User, for a Department Head approval, by right-clicking the Activity header, then selecting a desired condition. Common conditions include Approved, Rejected, Always, Skip and Error.

-- When might you add a condition? What would happen if the Dept. Head can't be found? In that case, the Activity would return a null value, which would then skip the Approval and use the first condition to transition to the next activity, such as Approved for example. You could add a Skipped transition to control what happens next.

**Stages** show Workflow progress, but work a little bit differently for Workflows that run against **Service Catalog Request Items [sc\_req\_item]**.

-- Workflows can provide a summary of Workflow progress by updating any field designated as a Stage field. For example, the **Incident [incident]** table has a State field that indicates progress, whereas Service Catalog uses the Stage field.

## Workflow States

servicenow

### Checked Out

The Workflow is available to run only for the user who has it checked out

### Unpublished

Workflow version that is no longer available for new contexts, but may be required for already running contexts (Published = false)

### Published

Workflow version that is available to all users and the platform that meet the conditions of the Workflow



servicenow

The **version** is the published model of the Workflow. The published version is available to the platform to use for creating new Workflow contexts.

**NOTE:** There can only be one published version of a Workflow at a time and past versions are kept as unpublished so there is a historical record. Additionally, executing Workflows will not be affected when a new version is published.

There are four distinct table interactions and four state changes that impact tables.

**NOTE:** When a Workflow version record is updated and the State changes to **published**, a table insert action occurs and adds the version record to the platform's current Update Set. Only published Workflows are captured in Update Sets.

## Section Summary

- What is a Workflow?
- Using Workflows
- Starting Workflows
- Models, Versions, Activities, Conditions, and Stages
- Workflow States
- Workflow Editors



### Lab 4.3 – Workflows:

- Modify group records
- Copy and edit the Service Catalog Item Request Workflow to create a custom Infinity Item Request workflow
- Validate and publish the workflow
- Test the workflow
- View the workflow history

**Lab Dependency** – This lab requires the completion of Labs 2.1 and 4.2.

# Lab Topics

- Modify group records
- Copy and edit the Service Catalog Item Request Workflow to create a custom Infinity Item Request workflow
- Validate and publish the workflow
- Test the workflow
- View the workflow history

**Lab Dependency** – This lab requires the completion of Labs 2.1 and 4.2.

## Lab 4.3 Workflows

Now that Cloud Dimensions has opened up enrollment for testing the Infinity across the entire organization and the Infinity has been created in the Service Catalog, it is time to implement procurement automation through a workflow.

There is a baseline workflow, **Service Catalog Item Request**, that the system administrator will copy and modify to fit the processes in place at Cloud Dimensions, as it relates to order fulfillment and deployment of the Infinity to employees.

### Copy and Rename Service Catalog Item Request Workflow Template

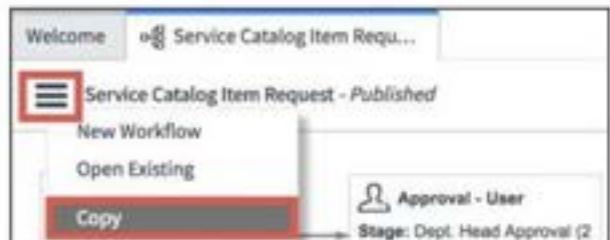
1. Workflow > Workflow Editor.

**NOTE:** The Workflow Editor displays in a separate tab or browser window.

2. From the Workflows tab on the right, select **Service Catalog Item Request**:

Name	Type	Updated by	Updated	Publisher
Service Catalog Item Request	Requested Item (sc_req_item)	glademaint	2019-01-14 09:49:52	true
Knowledge - Instant Publish	Knowledge (kb_knowledge)	admin	2014-11-05 11:52:29	true

3. From the Workflow Actions menu, select **Copy**:



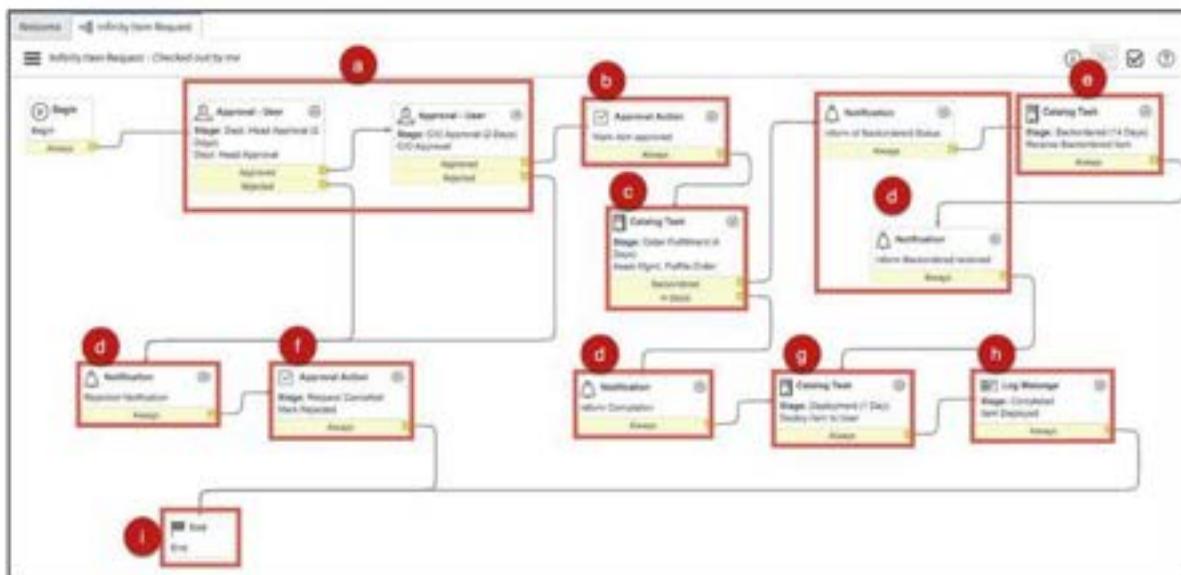
4. Name your workflow **Infinity Item Request**.

5. Click **OK**.

Verify the workflow tab displays the title **Infinity Item Request**:

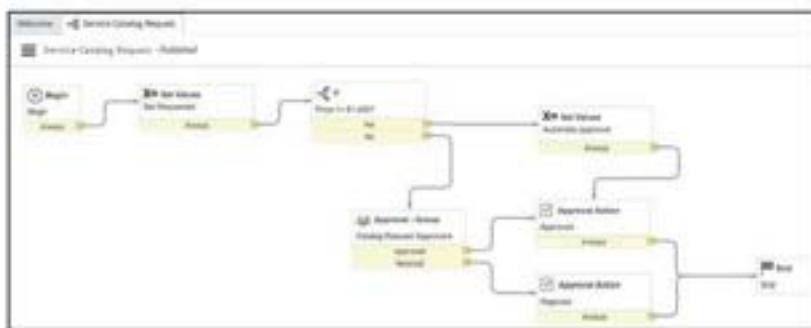


## Workflow Overview (What the Checked Out Workflow Looks Like):



- Sequential Approval Tasks
- Set the Approval field to Approved
- Assigned to the Procurement group to backorder or procure
- Notify the Requested for user
- Assigned to the Procurement group to check if backorder was received
- Set Approval field to Rejected
- Assigned to the Field Services group to deliver
- Writes to the Workflow Log
- Automated workflow stop

**NOTE:** All new Service Catalog requests have an initial approval rule that is evaluated by a separate Workflow, **Service Catalog Request**, shown here:



If the request amount is under \$1,000 USD it is automatically approved. This automated approval then instantiates any Workflow associated to requested items of the request.

In the **Service Catalog Item Request Workflow**, which you copied and will now modify in this lab, there are two User Approval Activities: Department Head and CIO. These are approval tasks for the item the Workflow is associated with (Infinity for example). A successful Department Head approval then triggers the CIO approval which, when successful, triggers the next Workflow activity, Mark item approved, and so on.

When the item is marked approved, then the first Catalog Task, Order Fulfillment, triggers. You will customize the Order Fulfillment activity in the lab. If the Infinity is in stock, a notification is sent. The second Catalog Task, Deployment, will end the Workflow after the task is completed successfully.

## Customize the Workflow

Begin by customizing the two Catalog Tasks: add a description of the steps involved for Order Fulfillment and Deployment, define the groups responsible, and customize the names to align with Cloud Dimensions processes.

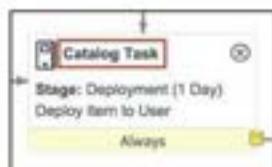
1. Double-click the Catalog Task with the Stage: **Order Fulfillment (4 Days)** and the Name **Asset Mgmt. Fulfils Order**:



2. Modify the Name to: **Infinity Order Fulfillment**.
3. Under the **Populate task variables** section, change the **Fulfillment group** to **Infinity Support**.
4. Modify the **Short description** field, overwriting existing values with the following values: **Tasks to fulfill an Infinity**
5. Click **Update**.

**NOTE:** Remember, Ted Keppel was previously added to the **Infinity Support** group; therefore, he will be impersonated later in the lab in order to complete this task.

- Double-click the Catalog Task with the Stage: Deployment (1 Day) and the Name Deploy Item to User:



- Modify the Name to: **Infinity Delivery**.
- Change the Fulfilment group to **Infinity Support**.
- Modify the form fields, overwriting existing values with Infinity values shown below:

Short description: **Deliver Infinity**

Instructions: **Deliver Infinity to requester or requester's manager.**

- Click **Update**.

**NOTE:** Remember, Beth Anglin was previously added to the **Infinity Support** group, so she will be impersonated later in the lab to complete this task.

## Validate and Publish the Infinity Item Request Workflow

- Validate the Workflow by clicking the **Validate** icon (Checkmark).

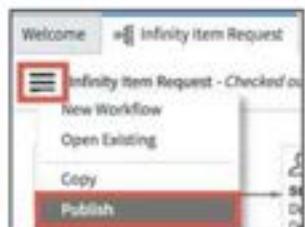


- A Workflow Validation Report displays.

**NOTE:** If you have warnings or critical errors, correct them before continuing.

- Close out of the Workflow Validation Report window.

4. Next, publish the Workflow to make it available for other users to checkout; from the Workflow Actions menu (which was used to copy a Workflow before), select **Publish**:



5. The workflow state updates in the header to **Published**.

### Associate Workflow to Service Catalog Item

In this next step, associate the Infinity Item Request workflow to the Infinity Service Catalog item created in the last lab.

1. Close the **Workflow Editor** tab (or window) to return to the main ServiceNow interface.
2. **Service Catalog > Catalog Definitions > Maintain Items**.
3. Search for and open the **Infinity** catalog item.
4. In the **Workflow** field, select **Infinity Item Request**.
5. Click **Update**.

### Test Workflow – Order an Infinity

In this next step, you will impersonate **Kevin Edd** and request an Infinity from the Service Catalog.

1. Impersonate **Kevin Edd**.
2. **Self-Service > Service Catalog**.
3. Click the **Hardware Category**, then select **Infinity**.
4. You can request either **color**, and, if desired, additional **memory**.

**NOTE:** If your order is greater than \$1,000 USD additional approvals (not documented here) are required.

5. Click the **Order Now** button to make the request.

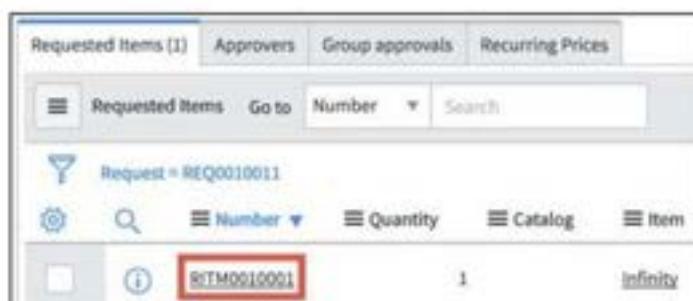
**NOTE:** The Infinity Item Request Workflow is initiated.

6. Verification that the request was submitted is displayed at the top of the form.
7. Record the Request number here: **REQ\_\_\_\_\_**.
8. Click the Infinity Description and record the Requested Item number here:  
**RITM\_\_\_\_\_**.

### Test the Workflow – User Approval

Next, review the progress of the Infinity Item Request Workflow, then approve the Department Head and CIO User Approval activities by impersonating the System Administrator. You could impersonate the individual users but this is quicker.

1. Impersonate System Administrator.
2. Service Catalog > Open Records > Requests.
3. Open the Infinity Request created with Kevin Edd's account.
4. In the Requested Items tab, click on the Requested Item Number (**RITMnnnnnnn**):



The screenshot shows a table with a single row of data. The columns are: Requested Items (1), Approvers, Group approvals, Recurring Prices. The first column has a sub-header 'Requested Items' with a dropdown arrow. Below the table, there are filter icons for Number, Quantity, Catalog, and Item. The 'Number' filter is set to 'RITM0010001'. The 'Quantity' field shows '1'. The 'Catalog' and 'Item' fields both show 'Infinity'.

5. Scroll down to Related Links and click **Show Workflow**.

This opens up summary of the running workflow context for the Infinity request:



Blue indicates activities that have been completed and green indicates the activity that the workflow is currently at.

6. Next, return to the main instance tab/window and select the **Approvers** tab on the **RITM** form.
7. Click **Requested** from the State column:

The screenshot shows the 'Approver' tab of the RITM form. The list view displays one record for 'Approval for = RITM0010001'. The columns are: Approver (checkbox), State (dropdown menu), and Approver (text field). The 'Requested' state is highlighted with a red border. The approver listed is 'David Loo'.

8. Click the **Approve** button to approve the request.

**NOTE:** You have approved this request as the system administrator, but for this workflow, you could also have impersonated David Loo to approve.

9. Now, repeat steps 6-8 to complete the second approval, CIO Approval, on Bow Ruggeri's behalf.

**TIP:** From the RITM Approvers tab, you can also right-click on the Requested record, then select **Approve** from the context menu as opposed to opening the approval record.

10. Impersonate Kevin Edd.

11. **Self-Service > My Requests.**

12. Open your Infinity Request.

13. Scroll down to the Requested Items section and expand the Stage field by clicking on the ">" icon:

The screenshot shows a table row for a requested item. The last column, labeled "Stage", contains a dropdown menu with the following options:

- Waiting for Approval (Request Approval)
- Dept. Head Approval - 2 Days (Dept. Head Approval - 2 Days (Completed))
- CFO Approval - 2 Days (CFO Approval - 2 Days (Completed))
- Order Fulfillment - 4 Days (Order Fulfillment - 4 Days (In progress))
- Deployment - 1 Day (Deployment - 1 Day (Pending - has not started))
- Completed (Completed Pending - has not started)

Notice the current Stage is set at **Order Fulfillment**.

### Test the Workflow – Complete Fulfillment Task

Next, you will impersonate Ted Keppel to approve the Order Fulfillment task since he was added as a member of the **Infinity Support** group and is responsible for completing the next phase in the Workflow.

1. Impersonate **Ted Keppel**.
2. **Service Desk > My Groups Work**.
3. Locate and open the Service Catalog Task associated with the Fulfillment workflow activity.

**HINT:** Search the Short description column for **Infinity**.

4. As **Ted Keppel**, assume the four items listed in the Description field are complete; click the **Close Task** button.

**[CHALLENGE]** View the progress of the Stage field for the Infinity request:

The dropdown menu for the Stage field lists the following stages with their descriptions and current status:

- Waiting for Approval (Request Approval)
- Dept. Head Approval - 2 Days (Dept. Head Approval - 2 Days (Completed))
- CFO Approval - 2 Days (CFO Approval - 2 Days (Completed))
- Order Fulfillment - 4 Days (Order Fulfillment - 4 Days (Completed))
- Deployment - 1 Day (Deployment - 1 Day (In progress))
- Completed (Completed Pending - has not started)

### Test the Workflow – Complete the Deployment

Since Beth Anglin was also added as a member of the **Infinity Support** group, she is responsible for delivering the Infinity once it is received.

1. Impersonate **Beth Anglin**.
2. **Service Desk > My Groups Work**.
3. Locate and open the Service Catalog Task associated with the Deployment workflow activity.

**HINT:** Search the Short description column for **Infinity**.

4. As **Beth Anglin**, assume you have delivered the Infinity; click the **Close Task** button.

## Test the Workflow – Verify Request Complete

1. Impersonate **System Administrator**.
2. **Service Catalog > Open Records > Requests**.
3. Confirm that the REQ is no longer in the list.

## FINAL VERIFICATION

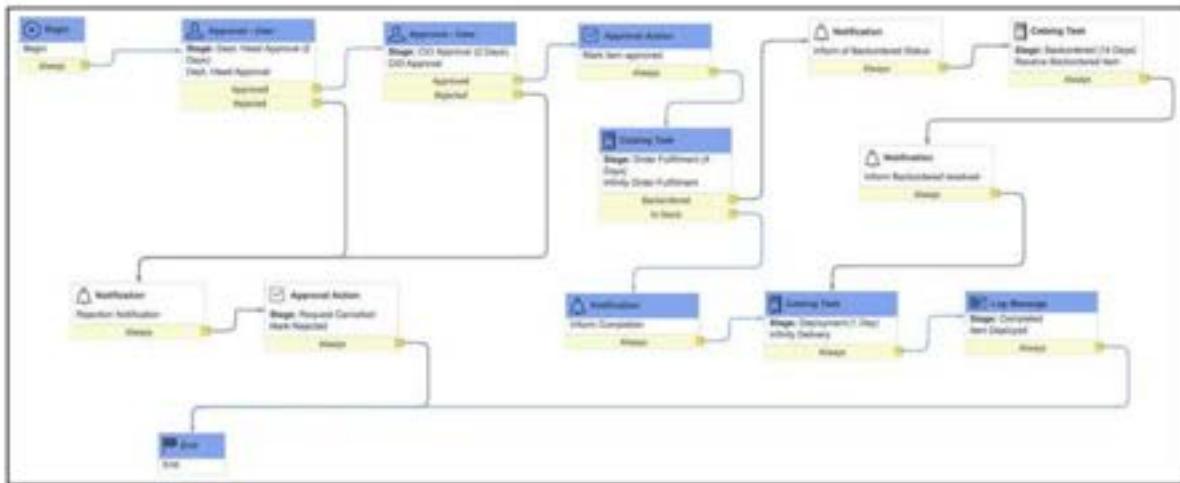
### View Workflow History

1. **Workflow > Live Workflows > History**.
  2. Click the **Ended** column header to date sort most recent at the top.
- NOTE:** The histories for all of the workflow steps of all requests are displayed in a single view. The list can be personalized to add other fields or a filter can be created to view only steps associated with a particular workflow or workflow activity.
3. Click the **Infinity Item Request** title in the **Context** column where the **Activity** is **Item Deployed**:

	Started	State	Ended	Context	Activity
1	2017-07-20 21:48:52	Finished	2017-07-20 21:48:50	Infinity Item Request	Infinity Delivery
2	2017-07-20 21:48:50	Finished	2017-07-20 21:48:50	Infinity Item Request	Item Deployed
3	2017-07-20 21:48:50	Finished	2017-07-20 21:48:50	Infinity Item Request	End

**NOTE:** This Workflow Context shows all of the workflow steps for the Infinity requested item that you just ordered, fulfilled, and delivered.

4. Select the **Workflow Activity History** tab and notice the nine Activities.
5. Under Related Links, click **Show Workflow** to show how the workflow executed your Infinity catalog item order:



*Congratulations, you have completed the Workflows Lab!*

## Module 4

service*now*

### 4.4 Service Level Agreements (SLAs)

- What is a Service Level Agreement?
- SLA Types
- SLA Components
- Conditions
- Default SLA Workflow

## What is a Service Level Agreement?

servicenow

A **Service Level Agreement (SLA)** is a record in the **SLA [contract\_sla]** table which defines a set amount of time for a Task to reach a certain condition, the table(s) to access, and what type of SLA is being evaluated

If the Task SLA does not reach the condition, the Task is marked **breached**

SLA includes actions that can be triggered at different times during the life cycle of the SLA

SLA's allow an IT service desk to track if their representatives are providing a specific level of service, and run reports on the success rates of the SLA actions.

For example, notify the manager when the SLA reaches 75% of its allotted time. Most commonly used to ensure incidents are being resolved within a certain amount of time.

## About Service Level Agreements (SLAs)

servicenow

SLAs Overview	Types of SLAs	Default SLA Workflow
There are several major components that work together to power the Service Level Agreements Plugin:	Service Level Agreement (SLA) Operational Level Agreement (OLA) Underpinning Contract (UC)	The Default SLA Workflow is designed to be used over and over
<ul style="list-style-type: none"><li>– SLA Definition</li><li>– Task SLA</li><li>– SLA Workflow</li><li>– SLA Automation</li><li>– SLA Conditions and Script Include</li></ul>		

### Overview: Key SLA Components

- **SLA Definition:** The record which defines the conditions that trigger the SLA
- **Task SLA:** The individual instances of the SLAs associated with particular tasks
- **SLA Workflow:** Workflow powers events or actions based on the SLA definition
- **SLA Automation:** The Business Rule and Scheduled Job that automate the SLA
- **SLA Conditions and Script Include:** A Script Include and reference record that can be used to customize the transitions between different SLA states

The **Task SLA [task\_sla]** table stores each of the individual SLAs attached to particular tasks. Unlimited SLAs can be running against a record. SLAs can only run against a table that extends the Task table.

**Types of SLAs:** While each type of Service Level Agreement may involve different stakeholders, their basic structure in the tool is the same, they track things you want tracked. The only difference between SLAs, OLAs, and Underpinning Contracts is the Type field on the Task SLA form. These SLA types basically all behave the same way. **NOTE:** Changing the Type field does not change the behavior of the Task SLA.

- An **Operational Level Agreement (OLA)** defines how departments work together to meet the service level requirements documented in an SLA.
- An **Underpinning Contract (UC)** is a type of SLA that defines and monitors the guarantees established with an outside supplier; it is a tool for supplier management.

## About Service Level Agreements (SLAs)

servicenow

Retroactive Start	Duration Type and Duration	Schedule and Time Zone
<b>Retroactive start:</b> When activated, works with the <b>Set start to</b> field and calculates the SLA start time  For example, an incident is received saying that email is down; an hour later the incident is updated with the email server being offline, but the start time is the same for both actions	<b>Duration Type:</b> The SLA performs the calculations and sets a day and time as the deadline for the SLA  <b>Duration:</b> When <b>User specified duration</b> is selected from the <b>Duration Type</b> list, an administrator can define the number of days and hours of the timer for the SLA	<b>Schedule:</b> Defines what is calculated as hours in a work day and number of days in a work week to use for the SLA  <b>Time Zone:</b> Specify the time zone for the SLA

### Important SLA Fields:

- **SLA Conditions:** You can set up to six SLA conditions: Start, Pause, Stop, Cancel, Resume, and Reset
- **Duration Type:** You can choose a specific duration from the list, or you can define your own. Typically, you would apply **User specified duration**.
- **Schedule:** Before you begin to look at how to calculate conditions for the scheduling of the SLAs, you can check to see what platform schedules have been set up for hours per work day and the number of days in the work week
- **Timezone:** The SLA definition's timezone is used when creating Task SLAs if the **Use the following time zone for SLA** property is selected in **Service Level Management > Properties > SLA Engine**. The timezone can be definition, schedule, location, or configuration item.

## SLA Conditions

service<sup>now</sup>

Start Condition	Pause Condition	Stop Condition
Enables you to define the conditions under which the SLA will be attached	Enables you to define the conditions under which the SLA will suspend	Enables you to define the conditions under which the SLA completes  If all of the specified stop conditions match, then the task SLA will complete regardless of whether it is breached
Cancel Condition	Resume Condition	Reset Condition
Enables you to define the conditions under which the SLA will cancel  You can specify the cancel conditions at the same time you specify the start conditions	Enables you to define the conditions under which the SLA will resume increasing elapsed time	Enables you to define the conditions under which the running SLA will be completed and a new SLA will be attached  For a new SLA to be attached, the start condition must match

**SLA Condition Evaluation** - Every task in the platform is evaluated in the following order:

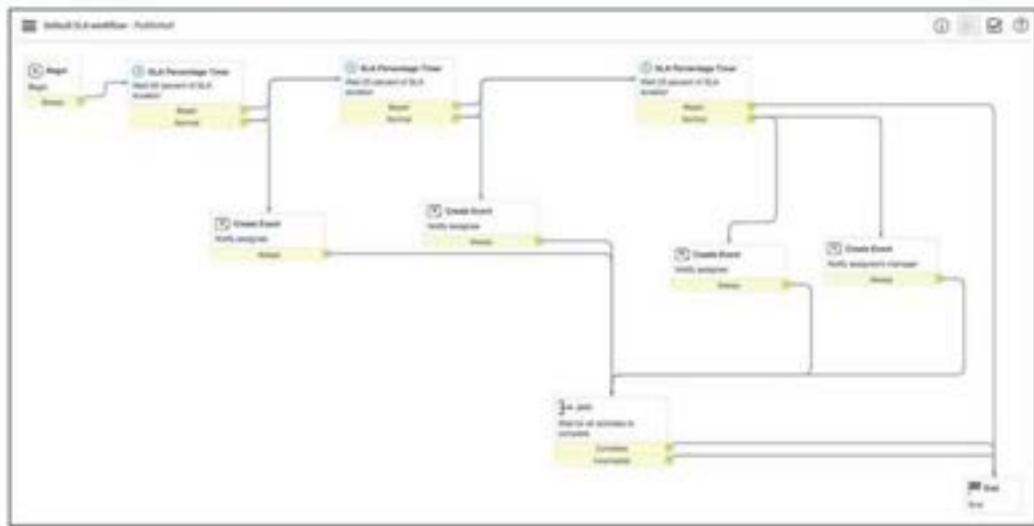
- Process new SLAs to determine if a new SLA record must be attached to a task
- Process existing SLA records attached to a task

SLA conditions are evaluated in the following ways:

- **Attach** if start condition matches and both the stop and cancel conditions don't match
- **Complete** if the stop condition matches
- **Pause** if the pause condition matches
- **Resume** if the pause condition doesn't match or resume condition matches
- **Reattach** if both the reset and the start conditions match
- **Cancel** if the start condition doesn't match or cancel conditions matches

## Default SLA Workflow

service*now*



The **Default SLA Workflow** is designed to be used with multiple service level agreements of any types. As you saw in an earlier module, ServiceNow also provides Workflows for Service Catalog Requests, Service Catalog Item Requests, and other types of Workflows, such as Routine Change and Emergency Change.

### Section Summary

- What is a Service Level Agreement?
- SLA Types
- SLA Components
- Conditions
- Default SLA Workflow



#### **Lab 4.4 – Service Level Agreements:**

- Create an SLA for Infinity security incidents
- Test the SLA

**Lab Dependency** – This lab requires the completion of Lab 1.3.

# Lab Topics

- Create an SLA for Infinity security incidents
- Test the SLA

**Lab Dependency** – This lab requires the completion of Lab 1.3.

Buster Wubbel and Winnie Reich have agreed upon a reasonable duration for how long it should take to resolve all Infinity security incidents.

They have asked the System Administrator to create a new SLA to let them test and finalize for production.

## Lab 4.4 Service Level Agreements

### Define an SLA for Infinity Security Incidents

1. Service Level Management > SLA > SLA Definitions.
2. Click New.
3. Complete the form as shown:

Name: **Infinity Security Incident Resolution**

Table: **Incident [incident]**

Duration: **Days 00 Hours 08:00:00**

Schedule: **8-5 weekdays**

Your form should look like this:

The screenshot shows the 'SLA Definition' form in ServiceNow. The 'Name' field is set to 'Infinity Security Incident Resolution'. The 'Type' field is set to 'SLA'. The 'Target' field is empty. The 'Table' field is set to 'Incident [incident]'. The 'Workflow' field is set to 'Default SLA workflow'. The 'Active' checkbox is checked. The 'Enable logging' checkbox is unchecked. The 'Application' field is set to 'Global'. The 'Duration type' is set to 'User specified duration'. The 'Duration' is set to 'Days 00 Hours 08:00:00'. The 'Schedule source' is set to 'SLA definition'. The 'Schedule' is set to '8-5 weekdays'. The 'Timezone source' is set to 'The caller's time zone'.

4. Add the Start conditions as shown:

Start Condition:  
**Active | is | true AND**  
**Category | is | Infinity AND**  
**Subcategory | is | Security**

The screenshot shows a configuration interface for start conditions. At the top, tabs for 'Start condition', 'Pause condition', 'Stop condition', and 'Reset condition' are visible. The 'Start condition' tab is selected. Below it, there's a section titled 'Start condition (1)' with buttons for 'Add Filter Condition' and 'Add OR Clause'. A note says 'All of these conditions must be met'. Three conditions are listed: 'Active' is set to 'is' and 'true'; 'Category' is set to 'is' and 'Infinity'; and 'Subcategory' is set to 'is' and 'Security'. Each condition has a delete button on the right.

5. Select the **Retroactive start** checkbox.

**NOTE:** Doing so will display the **Set start to** field.

6. Choose **Created** for the **Set start to** field.

7. Change the value of the **When to cancel** field to **Start conditions are not met**:

The screenshot shows a dropdown menu for 'When to cancel'. The option 'Start conditions are not met' is selected and highlighted in blue. Other options like 'When state changes' and 'When resolved' are also visible.

8. Select the **Stop condition** tab.

9. Add the following **Stop condition**:

**State | is | Closed OR**  
**State | is | Resolved**

The screenshot shows a configuration interface for stop conditions. The 'Stop condition' tab is selected. Below it, there's a section titled 'Stop condition (1)' with buttons for 'Add Filter Condition' and 'Add OR Clause'. A note says 'Any of these conditions can cause the SLA to end'. Two conditions are listed: 'State' is set to 'is' and 'Closed'; and 'State' is set to 'is' and 'Resolved'. Each condition has a delete button on the right.

**NOTE:** This SLA will take effect for any incident submitted with a category of Infinity and a subcategory of Security, and it will track time until the incident reaches the state of Closed or Resolved.

10. Click **Submit**.

## Test the Service Level Agreement

1. Impersonate Winnie Reich
2. Incident > Create New.
3. Fill out the form as follows:

Caller: Winnie Reich  
Category: Infinity  
Subcategory: Security  
Assignment group: Infinity Support  
Short description: Testing INC Security SLA

4. Save.
5. Scroll down to the Task SLAs section.

Notice 2 Task SLAs have been triggered:

Task SLAs (2)					
Task SLAs		Go to	SLA definition	Search	
Task = INC0010021					
SLA definition	Type	Target	Stage	Business time left	
<input type="checkbox"/>	Infinity Security Incident Resolution	SLA	In progress	8 Hours	
<input type="checkbox"/>	Priority 5 response (40 hours)	SLA	Response	In progress	1 Day 16 Hours

6. Scroll back to the top of the form and update the State field to Resolved.
7. Save the record.
8. From the Task SLAs section, hover over the Preview icon (circle with an "i") for the **Infinity Security Incident Resolution** record:



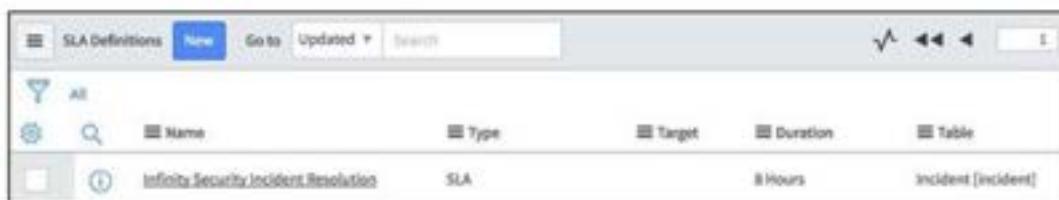
9. Click this icon to open the SLA context record.

10. Under **Related Links**, click **Show Workflow**.

**NOTE:** This Workflow context shows the lifecycle of the SLA for the incident.

## LAB VERIFICATION

### Service Level Agreement Definition



	Name	Type	Target	Duration	Table
<input type="checkbox"/>	Infinity Security Incident Resolution	SLA		8 Hours	Incident [incident]

*Congratulations, you have completed the Service Level Agreement lab!*

## Module Recap: Service Automation

servicenow

**Why** would you use these capabilities?

**When** would you use these capabilities?

**How often** would you use these capabilities?

Knowledge Management	Service Catalog	Workflows	Service Level Agreement
Knowledge Base	Service Catalog	Workflow Activities	SLAs
Publish Workflows	Variable/Variable Sets	Conditions	OLAs
Flag Article	Order Guide	Workflow States	UCs
User Criteria		Workflow Editor	Default SLA Workflow

service**now**

- 1
- 2
- 3
- 4
- 5

## Module 5

### Introduction to Scripting and Development

- 5.1 Scripting
  - 5.2 Performance and Upgrades
  - 5.3 Development
-

## Module 5

service**now**

### 5.1 Scripting

- What is ServiceNow Scripting?
- Client/Server Side Scripting
- Script Types
  - UI/Data Policies
  - UI Actions
  - Client Scripts
  - Business Rules
- What are Plugins?

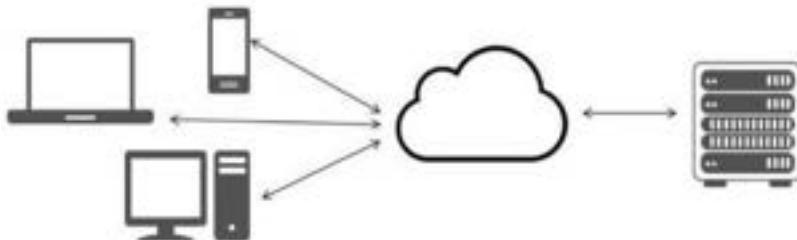
## What are Policies, Actions, Business Rules and Scripts?

servicenow

UI Policies, Data Policies, UI Actions, Business Rules and Scripts are all powerful tools for tailoring your ServiceNow instance and applications

**Client** refers to an application or system that accesses a remote service or another computer system, known as a server

**Server** is the computer program running as a service; a physical computer dedicated to running one or more services, or a system running a database



ServiceNow uses a Software as a Service (SaaS) model; the web browser is the client. Client to server round-trips take time and make the end-user wait for the round-trip to complete. The wait is caused because client side scripts are generally synchronous.

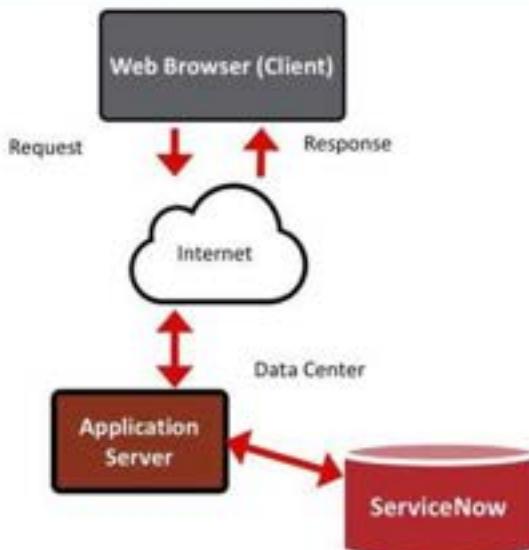
ServiceNow has over 30 places where code can be inserted to change the behavior of the platform. JavaScript is used almost everywhere and it is a very flexible and powerful language commonly known for its inclusion in most modern web browsers. This has made it almost mandatory for web development these days, with its simple syntax allowing many people to quickly add simple logic to web pages with minimum effort. Taking advantage of this familiarity, ServiceNow uses JavaScript both on the server and on the client.

You can learn more about the ServiceNow Scripting course at:

<http://www.servicenow.com/services/training-and-certification/scripting-in-servicenow-training.html>

## Basic Architecture Overview

service*now*



The web browser is the only thing that is installed on the client. The application server and the database live at the Data Center. Client scripts run on the client browser. Server scripts run on the server (which includes the database).

**NOTE:** Request + Response = Round trip.

## Sample Policies, Actions, Rules and Scripts

service<sup>now</sup>



Before you dive into detailed descriptions of each tool, let us look at an example of how they could be used together.

This example is from an Incident form, where:

- UI Policy can be applied to make the Number field read-only
- Data Policy can be applied to make the Caller field mandatory
- UI Action can be activated to make the Submit button visible only for users with appropriate roles
- Business Rules can trigger a confirmation email to go to the caller when an Incident is first submitted
- Client Script can be added to prompt the user to check the Knowledge Base before submitting an incident

## UI Policy and UI Policy Action, Data Policy and UI Action

servicenow

UI Policy / UI Policy Action (Client Side)	Data Policy (Client Side and Server Side)	UI Action (Client/Server Side)
<p>UI Policies offer an alternative to client scripts for dynamically changing information on a form</p> <p>Once you Save the UI Policy a new UI Policy Actions Related List displays</p> <p>Examples:</p> <ul style="list-style-type: none"><li>• Make fields read-only on a form</li><li>• Make a field mandatory</li><li>• Hide a field on a form</li></ul>	<p>Data Policies enable you to enforce data consistency by setting mandatory and read-only states for fields</p> <p>Examples:</p> <ul style="list-style-type: none"><li>• Enforce mandatory data on a form and on an import</li><li>• Enforce read-only data on a form and on an import</li></ul>	<p>UI Actions add buttons, links, and context menu items on lists and forms, making the UI more interactive, customizable, and specific to user activities</p> <p>Example:</p> <p>Make a Save button visible on form for appropriate users</p>

Creating a UI Policy Action gives further control by hiding or removing fields on the form if a condition is met. You can use UI Policies to define custom process flows for tasks.

Data Policies are similar to UI Policies, but UI Policies only apply to data entered on a form through a standard browser. Data Policies can apply rules to all data entered into the platform, including data brought in through Import Sets or Web Services and data entered through the mobile UI.

A Data Policy can make fields mandatory or set fields to read-only while a UI Policy Action can also hide a field.

UI Actions can be server or client side depending on the 'client' check box selection. UI Actions can contain scripts that define custom functionality.

## What is a UI Policy?

servicenow

A User Interface (UI) Policy is a form administration tool and is:

- A rule that applies to a form to dynamically change form information or the form itself
- An alternative to client scripts
- Run on the **client side** (on the browser)

Once you Save the UI Policy a new **UI Policy Actions** Related List displays

To immediately implement updates and changes to forms and lists, you can use UI Policies which allow you to add sophisticated controls without having to write scripts. To apply a UI Policy to all views, set **Global** to true.

Use a UI Policy to set fields on a form to:

- Optional or Mandatory
- Visible or Hidden
- Editable or Read-only

## What is a Data Policy?

servicenow

Create data policies to enforce consistency

Data Policy controls enable admins to set mandatory and read-only states for fields

Data Policies work at the server level so will apply to both lists and forms



A Data Policy enforces requirements on field and record data when the data is imported into ServiceNow or when the data in an Import Set is submitted through an external system. Data Policies can be opted out for Web Services and Import Sets. A Data Policy is used to set mandatory and read-only states on form fields. Data Policies can be used on lists to make a field read-only; the field will appear to be editable, but the update will fail.

The purpose of a Data Policy is to standardize the same data across ServiceNow applications. Data Policy controls are similar to UI Policy controls, but UI Policies are enforced only on data that passes through the browser and UI.

**NOTE:** UI and Data Policies are not about security, they are about managing the user experience.

## What is a UI Action?

servicenow

A UI Action is the method to put buttons, links, and context menu items on forms and lists, making the UI more interactive, customizable, and specific to user activities.

These buttons, links, and context menu items can be scripted, making them more powerful.

Buttons can be placed in the banner at the top of a list or at the bottom of a list using different controls

- To create a button that displays in the banner, personalize the UI Actions form to show the List banner button checkbox
- Use the List button checkbox to produce a button at the bottom of the list

### UI Actions include:

- Form buttons
- Form context menu items (right-click the header)
- Form links (Related Links in a form)
- List buttons
- List context menu items (right-click a record)
- List choices (at the bottom of a list)
- List links (Related Links at the bottom of a list)

**NOTE:** When the UI Actions Active box is checked, that means that the UI Action is running and visible, unless there is a condition met that specifies otherwise.

When Order 100 is specified, UI Actions with Order numbers greater than 100 will display after this UI Action, while UI Actions with Order numbers less than 100 will display before this UI Action, in the user interface.

## What is a Business Rule?

service*now*

A **Business Rule** is a piece of JavaScript configured to run when a record is displayed, inserted, updated, deleted, or when a table is queried.

A Business Rule can be set to run before or after the database action has occurred



Unlike UI Policies, Business Rules **do not** monitor fields on a form.

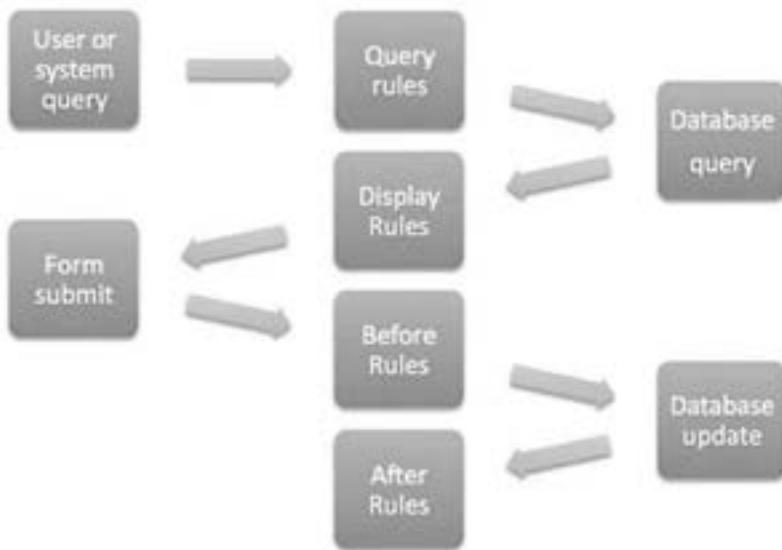
Although there are multiple ways to control behaviors in the ServiceNow application, most customization of platform behavior is done using Business Rules. Business Rules are loaded and initialized at the beginning of each interaction between a user and the platform.

Every Business Rule includes what table to run against and timing (before or after insert and more), what conditions to evaluate, what script to run based on the evaluation, and if it is client-callable.

Business Rules run on the server, but can be client-callable. If the **Client callable** setting is checked, the client can use AJAX to call the Business Rule.

## Business Rule Process Flow

service<sup>now</sup>



Business Rules are consistently applied to records regardless of how they are accessed-through forms, lists, or Web Services. This is one major difference between Business Rules and Client Scripts, which only apply when editing through the form.

Business Rules are **NOT** real-time:

- They do not monitor fields on a form
- They monitor records as they are inserted or updated

## When Do Business Rules Execute?

servicenow

The **When** setting lets you select when the Business Rule should execute:

- **before** a record is saved to the database
- **after** a record is saved to the database
- **async** (queued); Client and server work independently so the client is not waiting for the server
- **display** before the record is displayed (this can utilize a scratchpad area to access server data)

The primary objective of **display** is to use a shared scratchpad object, "g\_scratchpad", which is also sent to the client as part of the form. This is useful when you need to build client scripts that require server data that is not part of the record being displayed.

To learn more about this, you may want to consider attending the Scripting in ServiceNow course. You can learn more about it on the ServiceNow Corporate website:  
<http://www.servicenow.com/services/training-and-certification/scripting-in-servicenow-training.html>

## Client Scripts

servicenow

### What is a Client Script?

Client scripts are shipped to the client (the browser), and they run there instead of on the server



### Types of Client Scripts Supported

Client Scripts can execute in four ways:

1. onCellEdit
2. onChange
3. onLoad
4. onSubmit

Client Scripts are used for making “real-time” changes to the appearance of forms, or displaying different fields based on the value that is entered. Client Scripts allow for browser/form manipulation and verification such as making fields visible on a condition. An example of this would be an alert appearing when you change the priority of an incident. Client Scripts get executed on the browser. Run a Client Script when a database lookup is needed; if you think you need database info, and you need the info frequently (such as every form load) then ask: Is it a field you can add to the form but hide? Is it something you really, truly need?

Several types of scripts are supported:

- **onCellEdit()**: runs when a cell on a list changes value
- **onChange()**: runs when a particular field changes value
- **onLoad()**: runs when a form is loaded
- **onSubmit()**: runs when a form is submitted

Client Scripts can be defined to be triggered **onCellEdit**, to respond to use of the list editor.

Unlike **onLoad()** or **onSubmit()** scripts, **onChange()** scripts apply to a particular widget on a form, rather than to the form itself. They are fired when a particular value on screen changes. An **onLoad()** script runs when a form is first drawn and before control is given to the user to begin typing. Typically you use an **onLoad()** script to perform some client side manipulation of the document on screen. An **onSubmit()** script runs when a form is submitted. Typically you use an **onSubmit()** script to validate things on the form to make sure the submission makes sense. As such, **onSubmit()** scripts can potentially cancel a submission by returning false.

## Scripting Areas in ServiceNow

servicenow



ServiceNow Administrators can use JavaScript in many areas.

For more information on scripting, visit our product documentation site at [docs.servicenow.com](https://docs.servicenow.com)

## What are Plugins?

service<sup>now</sup>

Plugins also provide additional optional functionality that administrators can activate within a ServiceNow instance



Name	Version	Status	Description
ActivityScheduler	1.0.0	Active	com.glideui.activity.scheduler
AgentScheduler	1.0.0	Inactive	com.glideagent.scheduler
Asynchronous Web Services	1.0.0	Inactive	com.glideweb.services.aggregate
FileDevelopment	1.0.0	Inactive	com.glidefiledevelopment
FileDevelopment.LB	1.0.0	Inactive	com.glidefiledevelopment.lb
FileAttachment	1.0.0	Active	com.glideattachment
Instantaneous Connect Support	0.0.1	Inactive	com.glideconnect.instantaneous_support
GLAnalytics	1.0.0	Active	com.glideui.analytics
HRMaster	1.0.0	Active	com.glidehrmaster
Application Authorization	1.0.0	Inactive	com.glideui.app_authorization
Application Creator	1.0.0	In Active	com.glideui.app_creator

You have control over when to activate plugins.

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 first installing 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 checkbox. If the plugin depends on other plugins, these plugins and their activation status are listed.

Most plugins are published, and System Administrators can activate any published plugin. But, some plugins are available only by request due to operational considerations making the plugin only appropriate for certain deployments. In these cases, to activate the plugin, make a Service Catalog request to ServiceNow Technical Support using the Request Plugin Activation form at <https://hi.service-now.com>.

## Section Summary

- What is ServiceNow Scripting?
- Client/Server Side Scripting
- UI/Data Policies
- UI Actions
- Client Scripts
- Business Rules
- What are Plugins?



### Lab 5.1 – Scripting:

- Activate the time card management plugin
- Modify the Infinity list view
- Submit time card hours for an incident
- Create a UI Policy and UI Policy Action
- Create a Client Script
- Create a Business Rule

**Lab Dependency** – This lab requires the completion of Lab 1.3.

# Lab Topics

- Activate the time card management plugin
- Modify the Infinity list view
- Submit time card hours for an incident
- Create a UI Policy and UI Policy Action
- Create a Client Script
- Create a Business Rule

**Lab Dependency** – This lab requires the completion of Lab 1.3.

Cloud Dimensions has configured ServiceNow as much as possible by using baseline functionality and available features. Now, there are some configurations that will require additional functionality and require JavaScript.

The time card plugin will be activated to improve Infinity support efficiency, allowing support agents to input time spent working an incident.

Scripts will also be added to the platform to require and automate data entry, and improve user experience by notifying users of successful incident submissions.

## Activate a Plugin

1. Ensure you are logged into the instance as **System Administrator**.
2. **System Definition > Plugins**.
3. Search for and open the **Time card management** plugin record.

**NOTE:** The **Time card management** plugin is complimentary and available in the baseline ServiceNow instance. Click the **Help** link to learn what the plugin does.

4. Under Related Links, click **Activate/Upgrade**.
5. On the Activate Plugin pop-up, click **Activate**.
6. Once complete, from the Plugin Activation pop-up, click **Close & Reload Form**.

## Lab 5.1 Scripting

## Configure the Infinity Incident Form

1. **Incident > Open.**

Locate and open the incident Kevin Edd created to test the P1 Security Notification:

INC0010040	● 1 - Critical	New	Kevin.Edd	Infinity	Security	Testing P1 Security Notification
------------	----------------	-----	-----------	----------	----------	----------------------------------

**NOTE:** Your incident number may be different.

2. From the Form Control Menu, select **Configure > Related Lists**.
3. Add **Time Card->Task** to the Selected List after **Task SLA->Task**.
4. **Save** to return to the form.

## Test Submitting Time Card Hours

1. Update the incident record's State to **Resolved**.
2. **Save**.
3. Select the **Time Cards** tab, then click **New**:

Task SLAs (2)	Time Cards	Affected CIs
≡	Time Cards	<b>New</b>
≡	Go to	Wed

4. Input **3** into the **Wednesday** field:

≡ Monday	0
≡ Tuesday	0
≡ Wednesday	<b>3</b>

5. **Submit**.

Notice a new time card record appears under the **Time Cards** tab for the incident:

≡ Week starts on	≡ Category	≡ User	≡ State	≡ Sunday	≡ Monday	≡ Tuesday	≡ Wednesday
2017-07-19	Task work	System Administrator	Pending	0	0	0	<b>3</b>

6. Click **Update** to leave the incident record.

## Create a UI Policy

One requirement for Cloud Dimensions' process improvement is to require important Infinity support data, with the goal to speed up resolution time. To achieve this, a UI Policy and UI Policy Action will be created to make the **Assigned to** field required for internal (employee) Infinity incidents.

1. **Incident > Create New.**
2. From the Form Control Menu, select **Configure > UI Policies**.
3. Click **New**.
4. Fill out the UI Policy form information as shown:

**Table: Incident [Incident] (already selected)**

**Short description: Mandatory Assigned to if Employee = True & Category = Infinity**

Under the **When to Apply** tab,  
Conditions:  
**Employee | is | True AND  
Category | is | Infinity**

5. **Save.**

You have now defined a UI Policy and set the conditions on which it will be enforced. Next, indicate which field to affect by the UI Policy.

## Create a UI Policy Action

1. Scroll down to the **UI Policy Actions** section, then click **New**.
2. Enter the following information on the **UI Policy Action** form:

**Field name: Assigned to**  
**Mandatory: True**

3. Click **Submit** to save the UI Policy Action.

**NOTE:** A message may display indicating multiple UI Policies with the same order for this field exist. Basically, this means there are other policies that may run at the same time as this one, and that the end-result may not be predictable. You can view the other policies by selecting their names in the message – adjusting either policies accordingly.

### Confirm New UI Policy is Working

1. **Incident > Create New.**
2. Make sure the Employee field is on the form.

**NOTE:** If you do not see the Employee field, switch to the **Infinity** form view.

3. Select the Employee field.
4. Change Category to **Infinity**.

Notice that the Assigned to field is now mandatory.

5. Change Category to **Inquiry / Help** and notice that the Assigned to field is no longer mandatory.

### Create a Client Script

Continuing our process improvement, create a Client Script that will automatically populate the assignment group if an Infinity incident has been identified as internal (employee) or external (partner).

**NOTE:** You should have downloaded the **infinity-clientscript.txt** file which includes the JavaScript syntax to complete this portion of the lab.

1. From the current incident record, open the Form Control Menu.
2. Select **Configure > Client Scripts**.

**NOTE:** Leave the site/page if prompted.

3. Click **New**.
4. Fill out the form with the following information:

**Name: Set Assignment Group for Infinity**

**Type: onChange**

**Field name (this field appears after selecting a type): Subcategory**

**NOTE:** With an **onChange** Client Script type, ServiceNow is being told to execute JavaScript when a value in a field on the form changes. In this example, the Subcategory field is set to be “watched” by ServiceNow. If this field’s value changes, the Client Script’s Script will execute.

**Description: Set Assignment Group based on Employee Checkbox Value and Subcategory.**

5. The **Script** field was automatically populated with syntax based on the selection made in the **Type** field. Replace this text with the content of the file you downloaded:

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {  
    if (isLoading || newValue === '') {  
        return;  
    }  
    if(g_form.getValue('category')=='Infinity' && g_form.getValue('u_employee')=='true'){  
        g_form.setValue('assignment_group','Infinity Support');  
    }  
    if(g_form.getValue('category')=='Infinity' && g_form.getValue('u_employee')=='false'){  
        g_form.setValue('assignment_group','Infinity Customer Support');  
    }  
}
```

6. Click **Submit**.

This Client Script will check the values of the Employee checkbox, Category, and Subcategory fields.

If the Employee checkbox is selected, the category is Infinity, and a subcategory is selected, then the Assignment group will automatically be set to Infinity Support (responsible for handling internal support).

If the Employee checkbox is **not** selected, the category is Infinity, and a subcategory is selected, then the Assignment group will automatically be set to Infinity Customer Support (responsible for handling external support).

## Test Your Client Script

1. **Incident > Create New.**
2. Fill out the form as shown, in the exact order listed:

Caller: Beth Anglin  
Employee: [checked]  
Category: Infinity  
Short description: Did this incident submit?

The **Assigned to** field becomes mandatory (indicated by a red asterisk):

A screenshot of a user interface showing a search or assignment screen. On the left, there is a label 'Assigned to' with a red asterisk (\*) next to it, indicating it is a required field. To its right is a text input field containing a placeholder 'Search'. To the far right of the input field is a magnifying glass icon, typically used for search functions.

What caused this?

3. Set the **Subcategory** to **Request**.

Doing so sets the **Assignment group** field to **Infinity Support** because of the Client Script.

4. Uncheck the **Employee** checkbox.
5. Change the **Subcategory** field to **Network**.

Notice the **Assignment group** field value has changed to **Infinity Customer Support** because of the Client Script.

**NOTE:** If you did not get these results, debug your code and retest.

## Create a Business Rule

In this section of the lab, you create a Business Rule to display an alert, "Your incident has been successfully submitted" to all users who submit an incident.

1. From the Form Control Menu, select **Configure > Business Rules**.

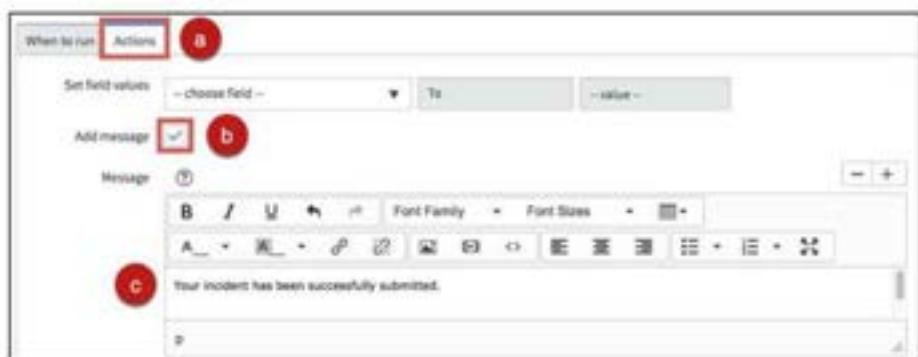
**NOTE:** If you left the incident record to debug and retest the Client Script, open a new incident record by navigating to **Incident > Create New**.

2. Click **New**.
3. Fill out the Business Rule form as shown:

Name: **Alert - Incident Submitted**  
Insert: **[check]**

4. Add a message:

- Click the **Actions** tab
- Check **Add message**
- Add message text: **Your incident has been successfully submitted.**



5. Click **Submit**.

### Test Your Business Rule

- Incident > Create New.**
- Fill out the incident form, including values for all mandatory fields.
- TIP:** Use the **Default Infinity Employee Incident** template.
- Click Submit.**
- The new Business Rule displays your message in blue on top of the list:

Incidents (Infinity view)						
		New	Go to	Number	Search	
Your incident has been successfully submitted.						
All	Active	True				
Number	Priority	State	Caller	Category	Subcategory	Short description
INC0010093	S - Planning	New	Beth.Anglo	Infinity	Network	Did this incident submit?

## LAB VERIFICATION

### UI Policy

Table: Incident [incident] Application: Global Active:

Short description: Mandatory Assigned to if Employee = True & Category = Infinity

When to Apply

Conditions: Add Filter Condition Add "OR" Clause  
All of these conditions must be met.

Employee	is	True	AND	OR	X
Category	is	Infinity	AND	OR	X

UI Policy Actions New Search for text Search

UI policy = Mandatory Assigned to if CDE = True, Category = Infinity, and Short Description Contains [cdit]

Field name	Mandatory	Visible	Read only
assignment_group	True	Leave alone	Leave alone

### Business Rule

Name: Alert - Incident Submitted Application: Global Active:  Advanced:

Table: Incident [incident] When to run: Actions

Insert:  Update:

Message: Your incident has been successfully submitted.

## Client Script

Name	Set Assignment Group for Infinity	Application	Global
Table	Incident [Incident]	Active	<input checked="" type="checkbox"/>
UI Type	Desktop	Inherited	<input type="checkbox"/>
Type	onChange	Global	<input checked="" type="checkbox"/>
Field name	Subcategory		
Description	Set Assignment Group based on Employee Checkbox Value and Subcategory		
Messages			
Script	<pre>1+ function onChange(control, oldValue, newValue, isLoading, isTemplate) { 2+   if (!isLoading    newValue === '') { 3+     return; 4+   } 5+   if(g_form.getValue('category')=="Infinity" &amp;&amp; g_form.getValue('u_employee')=="true"){ 6+     g_form.setValue('assignment_group','Infinity Support'); 7+   } 8+   if(g_form.getValue('category')=="Infinity" &amp;&amp; g_form.getValue('u_employee')=="false"){ 9+     g_form.setValue('assignment_group','Infinity Customer Support'); 10+ } 11+ }</pre>		

**Congratulations, you have completed the Scripting Lab!**

## Module 5

service*now*

### 5.2 Performance and Upgrades

- Troubleshooting Performance
- Response Time Indicator
- Transaction Logs
- ServiceNow Release Cycle
- Quarterly Patching Program
- Upgrading ServiceNow
- Cloning an Instance

## Troubleshooting Performance

service*now*

The performance of a ServiceNow instance is made up of four components:

- Application Server response
- Network latency and throughput
- Browser rendering and parsing
- Instance Cache



### Performance Components:

- **Application Server response:** Time for the application server to process a request and render the resultant page
- **Network latency and throughput:** Time for the network to pass your request to the server and the response back
- **Browser rendering and parsing:** Time for your browser to render the HTML and parse/execute JavaScript
- **Instance Cache:** The amount of platform resources available for processing

## Assess Form Performance: Response Time Indicator

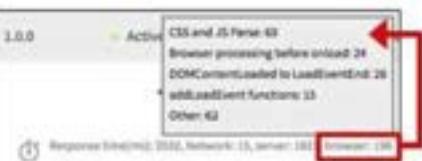
servicenow

### Use Response Time Indicator to look for:

- Poor list response times
- Poor form load and submit response times
- Poor module response times

### End-to-end response times:

- **Response time(ms):** Total time from the Create New click to the form load
- **Network:** Total time spent "traveling"
- **Server:** Total time spent processing the request on the server
- **Browser:** Total time spent rendering the form
  - Includes running client side script



The information provided by the Response Time Indicator is very useful when reviewing form performance. For example, you can toggle on the word "browser" to display and hide browser timings.

**NOTE:** To show/hide the response time indication, click the clock icon. Administrators can disable the response time by setting the `glide.ui.response_time` property to false.

### The Response Time Indicator:

- Displays on the bottom right of forms and lists once activated
- Is not available for the first transaction in sessions

### Look for:

- Inefficiencies: is there a Related Link that loads slower than the rest?
- Slow client scripts
- Bottlenecks in load time and address

## Transactions Logs

servicenow

### Transaction Log

The Transaction Log stores a record of all browser activity for an instance, and shows information about various activities including:

**Created:** Date and time of the transaction

**Created By:** The user who executed the transaction

**Response time:** Round trip response time for the browser request (in milliseconds)

### Client Transaction Log

Transactions (All user) versus Client Transactions:

The definition of what is a transaction in the Client Transaction Log is different than the definition of a transaction

In the Transactions (All user) Log which considers every redirect a transaction within the application.

It is easy to check application server and network responses: type **system logs** in the Filter navigator. Instances automatically log vital statistics of every transaction they process. That information is available to you as a System Administrator.

To look at the log, navigate to **System Logs > Transactions**. Reason to run Transaction Log:

- Good for finding slow running background jobs. **TIP:** You can Sort (z to a)
- Good for determining if a job ran or not

ServiceNow instances automatically log the vital statistics of every transaction processed, and that information is available to you as a System Administrator.

**NOTE:** You can limit the transaction list to display transactions that took place only during the time period you are investigating.

A login is a good example:

1. Type your credentials, then click login; the login records a transaction that attempts to authenticate you.
2. Once you are authenticated the platform redirects you (another transaction).
3. The platform then loads your homepage (another transaction).

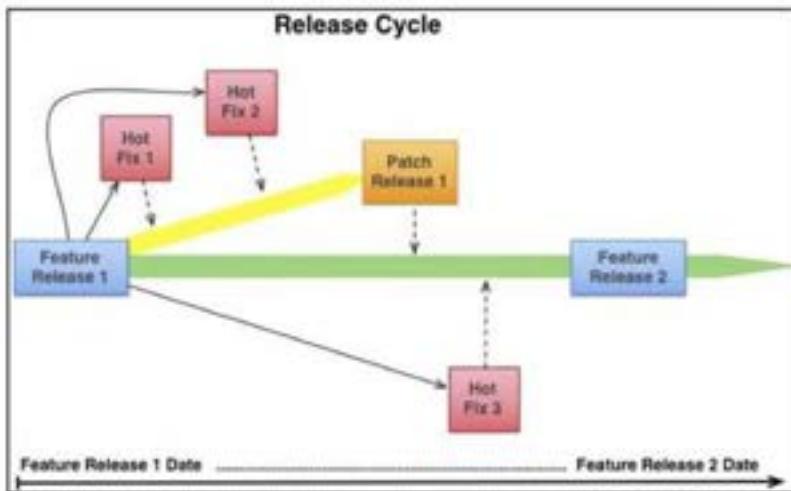
This is recorded as three transactions in the Transactions (All user) Log, but it is recorded as one transaction in Client Transactions Log.

## ServiceNow Release Cycle

servicenow

The ServiceNow release cycle is designed to provide optimal stability and quality, with the flexibility to quickly address problems and deliver new features

ServiceNow uses the concept of a family for a given feature set



A family contains:

- Feature releases, which contain new functionality and fixes to existing functionality
- Patch releases and hotfixes, which provide problem fixes and are released as needed

Customers are notified prior to the start of an upgrade rollout period. Release notes are available for all releases. Customer customizations are typically preserved throughout all upgrades.

ServiceNow releases are named after world cities, alphabetically, starting with Aspen, then Berlin, Calgary, Dublin, Eureka, Fuji, Geneva, Helsinki, Istanbul, etc.

## Release Terminology

service<sup>now</sup>

Type	Scope	Upgrade Policy
Feature release	<ul style="list-style-type: none"><li>Introduces new features*</li><li>Includes all available fixes to existing functionality</li><li>Is production-oriented; quality and stability are of the highest priority throughout the life cycle</li></ul>	<ul style="list-style-type: none"><li>Applied automatically during the rollout period</li></ul>
Patch release	<ul style="list-style-type: none"><li>Supports existing functionality with a collection of problem fixes</li><li>Includes all previously issued hotfixes for a given release</li><li>Does not include new features</li></ul>	<ul style="list-style-type: none"><li>Applied as needed on a per customer basis</li><li>Provided for the current and previous two feature releases</li></ul>
Hotfix	<ul style="list-style-type: none"><li>Supports existing functionality with a specific problem fix for a feature release</li><li>May or may not include any previous fixes for a given release</li><li>Does not include new features</li></ul>	<ul style="list-style-type: none"><li>Applied as needed on a per customer basis</li><li>Provided for the current and previous two feature releases</li></ul>

*\*A feature provides a complete solution that customers can implement to add value to their organization. New features are only available as part of a feature release. Features are supported with patches and hotfixes.*

### How to Communicate "No Upgrade Requests"

Customer may submit a Support Request for "No Upgrade" no fewer than five (5) business days' prior to a pending Upgrade of the ServiceNow Applications or the ServiceNow Automation Platform. Subject to the terms and conditions of this Upgrade Policy, Customer's "no Upgrade" request shall be granted, and the Upgrade shall not be applied to Customer's instances of the Subscription Service.

## Quarterly Patching Program (QPP)

servicenow

Quarterly	Release Families	Update Record
<p>The ServiceNow Quarterly Patching Program (QPP) patches customer instances on a quarterly scheduled interval throughout the year</p> <p>The goal of the program is to ensure that all customer instances have the latest security, performance, and availability updates</p>	<p>Instances on a supported release family will be patched to a target version within the same family</p> <p>QPP will not upgrade instances to a new family but rather patch them within family; the only caveat is if a customer is on an unsupported family, they will be upgraded a new family</p>	<p>ServiceNow has implemented an <b>Update Record</b> which includes two-way, private communication between the customer and ServiceNow to communicate any questions or comments about the QPP Program</p>

### How is the Quarterly Patching Program (QPP) different than an upgrade?

Unlike an upgrade, which is typically intended to provide enhanced or increased functionality, this program is focused on ensuring that customer instances are optimized for availability, performance, and security. An **upgrade** is the act of moving from one release family to another. **Patching** is moving up versions within a release family. The Quarterly Patching Program (QPP) is focused on in-family patching. While mandatory upgrades may be required, they will be done in a separate process, not part of Quarterly Patching. In the QPP, Customer, Partner and internal instances will be patched to the latest version within the same supported release family.

**NOTE:** Patches generally do not "break" an instance and do not require extensive testing.

Customers can access the Update Record on their Upgrade Management Dashboard on HI: [https://hi.service-now.com/upgrade\\_dashboard.do](https://hi.service-now.com/upgrade_dashboard.do). In the status field for each instance, there is a link to "Update Record". Although there is a link with each instance, the same record is used for all of the customer's instances. To check the status of a quarterly update for a specific customer or any of the communications with the customer about the quarterly updates, you can look up the version update record for the specific customer. In /HI, navigate to **Version Management > Companies > All Company Version Updates**.

## About Upgrades

servicenow

Upgrade History Record	Upgrades and Release Version
The Upgrade History module tracks every upgrade made to the platform	<b>Running Release:</b> System administrators can check which release is running on an instance: <b>System Diagnostics &gt; Stats &gt; Stats</b>
Each record that is examined as part of the upgrade is tracked	<b>Confirm Upgrade:</b> To verify that a recent upgrade to a feature release has been performed: <b>System Logs &gt; Events</b> then find the event <b>system.upgraded</b>
Information about what the platform actually decided to do with a particular record is kept	<b>New Build Notification:</b> You can configure your instance to send an email at the end of an upgrade

The Upgrade History module helps an administrator locate and resolve upgrade conflicts by reverting customizations to the base version, if necessary.

**Check Running Release:** Locate the Build name, Build date, and Build tag. Match the build name to the release name in the Release Notes. Match the build date and tag to the build date and tag in the Release Notes.

**Confirm Upgrade:** The event **system.upgraded** can trigger the System Upgraded notification. If the event has a State of error, the notification is not sent; however, the upgrade may have completed successfully. Navigate to **System Diagnostics > Upgrade Log** and locate the message "Notifying HI that upgrade has been completed." To confirm that a patch has been applied, find the **system.patched** event. You can create a notification for this event; it is not available by default.

**New Build Notification:** You can update the notification by adding the user you want to receive a notification email. Navigate to **System Policy > Notifications** and locate the email notification named **System Upgraded**.

## What is Upgraded and What is Not?

servicenow

- Customizations Exempt in Upgrades

### When an upgrade is applied

- Any record that has been altered from the base state by the customer will be **exempt** from the upgrade
- This prevents an upgrade from overwriting and breaking user customizations

### Customizations are tracked

- Any record that has an updated record in the **Customer Update [sys\_update\_xml]** table is skipped during the upgrade

Customer customizations are typically preserved by the upgrade process by the following mechanism: when an object is customized, a corresponding record is added in the **Customer Updates [sys\_update\_xml]** table. This table maintains the current version information for all objects that have been customized.

To prevent customizations from being overwritten or broken by platform upgrades, the upgrade process automatically skips changes to objects that have a current version in the Customer Updates table.

You may want to overwrite certain customizations when a software upgrade contains a feature that you would like to implement.

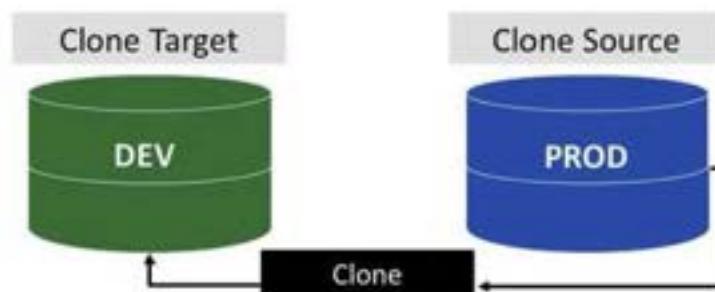
## Cloning an Instance

servicenow

**Cloning** overwrites the entire content on the target instance, including data and configurations

When should I clone my instance stack?

- Before starting a formal testing cycle
- After a comprehensive or long development cycle
- After the completion of an application upgrade



A clone does not have to be a total replacement of the existing instance; controls are provided to exclude applications and features from being affected by the cloning process as this could cause inconsistencies between instances. Also, not all system elements and properties are cloned ; for example, email settings, CSS, and instance colors.

Users with the **admin** or **clone\_admin** role use the System Clone application to schedule cloning without requiring a clone request on the ServiceNow UI system.

**System Clone > Clone Definition > Exclude Tables** lists the tables on the target system that are not overwritten by the clone process. When a table is excluded, an empty table is preserved. To keep the table data, navigate to **System Clone > Clone Definition > Preserve Data** and add the table to the list.

### Section Summary

- Troubleshooting Performance
- Response Time Indicator
- Transaction Logs
- ServiceNow Release Cycle
- Quarterly Patching Program
- Upgrading ServiceNow
- Cloning an Instance



### Lab 5.2 – Performance and Upgrades:

- Check current running release
- View the upgrade history
- Enable calculations and record baseline statistics

# Lab Topics

- Check current running release
- View the upgrade history
- Enable calculations and record baseline statistics

It is possible to monitor instance performance over time especially as more and more data, as well as script types, applications, and other functionality, is added to the database.

**NOTE:** ServiceNow monitors your instances for you.

## Lab 5.2 Performance and Upgrades

Cloud Dimensions understands this importance and has requested that the system administrator gather some baseline statistics to have as a reference point.

### Check Running Release

1. System Diagnostics > Stats > Stats.
2. Locate the Build name, Build date, and Build tag:

Statistics for: Demo Server @ [sfuir-007.lab.service-now.com:443](#) at: Thu Jul 20 20:19:58 PDT 2017 ([Refresh](#))  
Connected to cluster node: in-10-164-166-106.ec2.internal:labinstance001\_443  
Build name: Jakarta  
Build date: 05-23-2017\_1304  
Build tag: glide-jakarta-05-03-2017\_patch0-05-18-2017  
Environment name: sfuir-007

### View Upgrade History

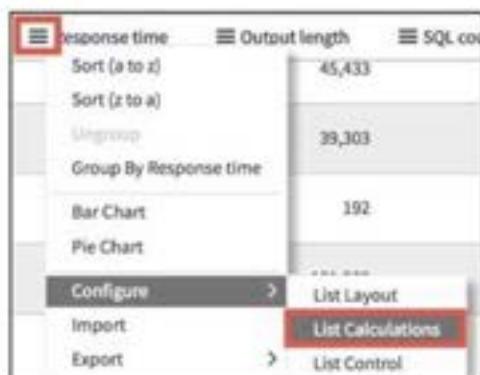
1. System Diagnostics > Upgrade History.
2. A record for every upgrade that has been run displays so you can track any changes and additions:

System Upgrades	Go to	Upgrade started	Search		
All	From	To	Upgrade started	Upgrade finished	
<input type="checkbox"/>		empty	glide-jakarta-05-03-2017_patch0-05-18-2...	2017-05-25 18:08:43	2017-05-25 18:39:12

## Access System Logs and Turn on List Calculations

You can turn on various list calculations for the most watched columns: SQL time, Client response time, Client network time and Browser time.

1. **System Logs > Transactions (All user).**
2. Open the List Control Menu for the **Response time** column, then select **Configure**, then finally select **List Calculations**:



3. From the calculations window, make sure the **Maximum value** and **Average value** checkboxes are selected.
4. Click **OK** to close the calculations window.
5. Repeat steps 2-4 for the **SQL time** column.
6. After the list reloads, scroll to the bottom to see the calculation values for the Response time and SQL time columns.

## Record Your Baseline Statistics for Transactions Time:

**System Logs > Transactions (All user)**

Avg Response time: \_\_\_\_\_

Avg SQL time: \_\_\_\_\_

## [Challenge] Turn on List Calculations in the Client Transactions Module

Add **Average** for **Client response time**.

Also add **Average** for **Client network time** and **Browser time**.

## **Record Baseline Statistics for Client Transactions Time:**

Avg Client response time: \_\_\_\_\_

Avg Client network time: \_\_\_\_\_

Avg Browser time: \_\_\_\_\_

## **Record Baseline Statistics About the Instance**

Monitor your ServiceNow instance using **Stats** which contains key metrics:

- Servlet Memory
- Available Semaphore Sets
- Response Time

1. **System Diagnostics > Stats > Stats.**

2. Locate the **Servlet Memory** section.

**Servlet Memory** is a useful problem indicator. Free percentage is typically between 10-60%. As long as you are not below 10 for an extended period of time you are in good shape.

**NOTE:** The **stats** page only shows a snapshot in time and does not necessarily indicate a prolonged issue. A sample of stats pages should be taken over a period to establish whether there is a persistent problem.

3. Look under **Servlet Memory** and record your values:

Max memory: \_\_\_\_\_ In use: \_\_\_\_\_

Allocated: \_\_\_\_\_ Free percentage: \_\_\_\_\_

**Discuss:** Find a partner and compare your values. Are they similar?

Free percentage should be over 20%. Refresh the page a few times and pay attention to the Free percentage. Is it repeatedly above 20%? [ ] Yes [ ] No

4. Locate the **Semaphore Sets** section.

**Semaphore Sets** show the number of semaphores in use by the selected node.

Semaphores serve as simultaneous transaction gates; the point of semaphores is to make sure you do not have too many things running at the same time and causing over consumption of resources by controlling the number of user transactions that can be run in parallel; specifying the maximum number of concurrent transactions

5. Record your Default Semaphore Sets values:

Available semaphores: \_\_\_\_\_

Maximum transaction concurrency: \_\_\_\_\_

Maximum concurrency achieved: \_\_\_\_\_

6. Finally, locate the sections that list **Server Response Time**, **Client Network Time**, and **Client Browser Time**.

Your application server is designed to run efficiently without significant tuning, and that usually turns out to be the case. However, there is a difference between "efficiently" and "optimally", and it is a rare deployment that can't benefit from some degree of tuning. So, it is useful to record baseline performance statistics:

	Server Response Time	Client Network Time	Client Browser Time
1 minute	440	005	
5 minute	404	006	
15 minute	392		
60 minute	388		
1 day			

**Congratulations, you have completed the Performance and Upgrades Lab!**

## Module 5

service**now**

### 5.3 Development

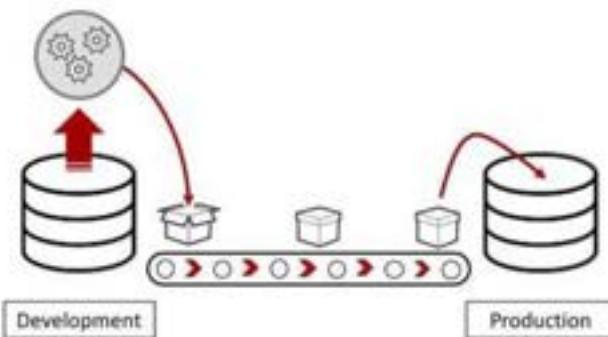
- What are Update Sets?
- Comparing, Reverting, and Merging Update Sets
- Application Scopes
- ServiceNow Studio
- Integrations
- Developer Documentation

## Update Sets

servicenow

An Update Set is a group of customizations that can be moved from one instance to another

Update Sets allow administrators to group a series of changes into a named set and then move them as a unit to another platform



An Update Set example:

- A set of enhancements to Incident Management can be grouped in an Update Set called Incident Management 2.0.
- While Incident Management 2.0 is marked as the current Update Set, all process changes are tracked in it.
- Once the Update Set is marked as complete, it is ready to be moved to a test or production instance.

Basically an Update Set record is a “point in time” XML snapshot of process records. An Update Set works by writing changes from tracked tables to the **Customer Update [sys\_update\_xml]** table.

An Update Set is used to apply changes that have been checked and verified in another instance. When merging multiple Update Sets, if several Update Sets have modified the same object, (for example: the Incident form), the most recent change will be the one moved to the new, merged Update Set.

An Update Set is a container for configuration records. By navigating to **System Update Sets > Local Update Sets**, you can create a new Update Set or set an existing one as your current Update Set. Use an Update Set to migrate your code. When an Update Set is completed, you can transfer it to another instance to move customizations from development, through testing, and into production.

## What is Captured in an Update Set?

servicenow



### Captured (Process Records)

- Business Rules
- Client Scripts
- Fields
- Forms and Form Sections
- Reports
- Tables
- Views
- Roles
- Workflows



### NOT Captured (Data)

- New Data Records
- Modified Data Records
- Tasks
- Modified Cls
- New Users and Groups
- Schedules
- Scheduled Jobs

### Homepages

Homepages must be manually added to an Update Set

1. Homepage Admin > Pages
2. Right-click a homepage record
3. Select Unload Portal Page

**NOTE:** Data is not captured in an Update Set. Examples: a new Incident or new Change Record would not be in an Update Set.

When completing work, you may want to move data records with your updates. These records can be useful for testing or training. Data (such as user records, Cls, or locations) can be moved using the Export XML function.

## Compare, Revert, and Merge Update Sets

servicenow

Comparing Objects	Revert Compared Versions	Merge Update Sets
Changes are tracked and you can customize objects on a table with the <code>update_synch</code> attribute:	You can compare versions before reverting a change	Update Sets can be merged
– Tables	You cannot revert database changes	If two users are working on separate Update Sets they can be combined into one Update Set for easy transfer
– Forms	You can only revert to the most recent base version	
– Fields		
– Business Rules		
– Client Scripts	You cannot revert to an older base version	
– Views		

You can compare a version to the current version for any customizable object that a user has modified, such as a form layout or Business rule. The **Update Versions** [`sys_update_version`] table supports this feature.

**NOTE:** Administrators can suppress versions for specific tables.

To revert changes:

1. View a list of versions for an object.
2. Right-click a version and select **Revert to this version**.
3. Click **OK** to confirm the action.
4. The selected version becomes the current version.

If both Update Sets have an update for the same object (for example, both Update Sets modify the Incident form), the most recent change will be the one moved to the new merged Update Set. The other update will be left in its original Update Set. Once a merge is performed, the other Update Sets remain, and if there were collisions, the duplicates remain where they were.

This provides a reference for what got moved and what did not. After merging and validating, it is a good idea to delete or empty the original Update Sets. The platform will not remove an update from an Update Set unless it was the one chosen for the merge.

## Plan and Manage Update Sets Process

service<sup>now</sup>

### Best Practices

#### Manage Changes and Communicate Effectively

- Have a plan & identify a common migration path
- Know what is being developed & make sure the Administrators are aware of developments

#### Include many changes in one set

- **Not:** “Many changes, many sets”
- Group like items in a small manageable set

Check that both instances are the same version since customizations may not work if they rely on code that has changed between versions.

Determine the changes to make in a single Update Set since ServiceNow recommends limiting Update Sets to a maximum of 100 records to reduce the number of potential conflicts and make it easier to identify and review changes.

Ensure that all platform records have matching `sys_id` fields since some platform records are created on an instance after provisioning and do not match between different instances, leading to problems with Update Sets. The best way to avoid this issue is to provision production and sub-production instances. You can clone the production instance onto the sub-production instance.

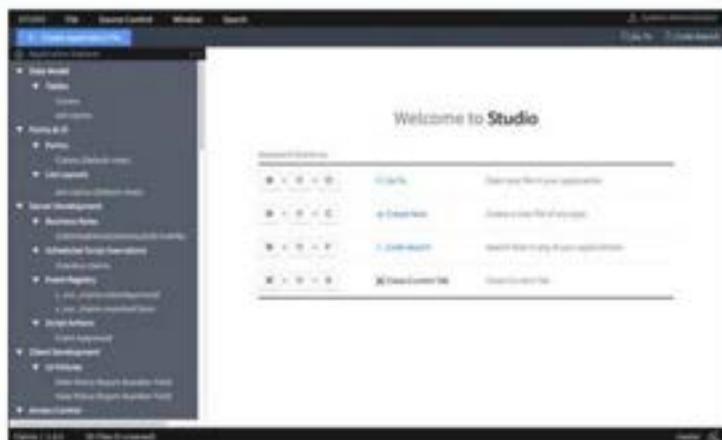
**NOTE:** Newest change will always overwrite older changes.

## ServiceNow Studio

servicenow

**ServiceNow Studio** provides an IDE-like interface for application developers to create custom applications

Application developers can also access Studio to import or open applications



ServiceNow Studio provides an IDE-like interface (integrated development environment). It offers a simple way to identify and interact with application files, create files as you develop, and modify existing application files in a tabbed environment. Accessing Studio requires an admin or a delegated developer role.

With Studio, application developers can:

- See exactly what files comprise their application in the **Application Explorer**
- Add new files to their application using a single **Create Application File** interface
- Navigate to files using familiar search-by-name or by-type behavior with the **Go To** dialog
- Find code both within and outside an application using the **Code Search** tool
- Operate on multiple files at once using the tabbed interface
- Operate on multiple applications at once using multiple studio windows
- Publish the application to company instances or the ServiceNow Store
- View information about the current application from the **Status Bar**

**NOTE:** Studio is not intended for global applications and can behave unexpectedly when editing them.

## Application Scopes

servicenow

Application scoping protects applications by identifying and restricting access to available files and data

Applications developed prior to application scoping are in the **global scope**

All custom applications have a **private scope** that uniquely identifies them and their associated artifacts

App A	App B
Table A	Table B
UI Policy A	UI Policy B
Business Rule A1	Business Rule B1
Business Rule A2	Business Rule B2

Administrators can specify what parts of an application are accessible to other applications from the custom application record and each application table record.

For example, suppose that you create a conference room booking application in its own application scope. By default, the application can access and change its own tables and business logic but other not applications unless you give them explicit permission.

The application scope ensures:

- The conference room booking application does not interrupt core business services
- Other applications do not interfere with its normal functioning

By default, all custom applications have a private scope that uniquely identifies them and their associated artifacts with a namespace identifier. The application scope prevents naming conflicts and allows the contextual development environment to determine what changes, if any, are permitted. Application developers specify an application scope when they create an application.

The global scope is a special application scope that identifies applications developed prior to application scoping or applications intended to be accessible to all other global applications.

The system adds a namespace identifier to the front of application artifacts such as tables, scripts, and configuration records.

## Integrations

service*now*

ServiceNow integrates with many third-party applications and data sources

Standard integrations for ServiceNow include:

- Login (Single Sign-On)
- LDAP
- Communications
- Monitoring
- Discovery & Systems Management



The most common processes required for integration are the CMDB, Incident Management, Problem Management, Change Management, User Administration, and Single Sign-On.

A variety of techniques can be used, most notably Web Services, LDAP, Excel, CSV and email, as well as any industry standard technologies that use SOAP or REST WSDLs.

## Documentation: Developer

servicenow

The screenshot shows the ServiceNow Developers website. At the top, there's a navigation bar with links for PROGRAM, LEARN, API, MESSAGE, COMMUNITY, EVENTS, and BLOG. On the right side of the header, there are links for PROFILE, LOG IN, and SIGN UP. Below the header, there's a large banner with the text "Build Apps Faster" and a subtext "All of the resources you need to build applications more rapidly than ever before". A "SIGN UP" button is located in the upper right corner of this banner. To the left of the banner, there's a photo of two people working together. Below the banner, there's a section titled "Develop Enterprise" with the subtext "Harness the strength of the ServiceNow development platform". To the right of this section, there's a detailed code example for a "GlideRecord" class:

```
class MyGlideRecord extends GlideRecord {  
    public MyGlideRecord() {  
        super("mytable");  
    }  
    public void init() {  
        // Initialize  
    }  
    public void validate() {  
        // Validate  
    }  
    public void insert() {  
        // Insert  
    }  
    public void update() {  
        // Update  
    }  
    public void delete() {  
        // Delete  
    }  
}
```

Below the code, there are sections for "GlideRecord" and "GlideRecord (String TableName)".

[developer.servicenow.com](http://developer.servicenow.com) is a great resource for developers, but also anyone interested in developing applications within ServiceNow.

There are great resources here related to development, including: scripting API references, free training and documentation, and access to a free, personal developer instance.

### Introduction to Scripting and Development

© 2017 ServiceNow, Inc. All Rights Reserved

### Section Summary

- What are Update Sets?
- Comparing, Reverting, and Merging
- Application Scopes
- ServiceNow Studio
- Integrations
- Developer Documentation



#### Lab 5.3 – Development:

- Review course work collected in an Update Set
- Sign up for a developer instance
- Retrieve an Update Set, committing platform changes to the developer instance
- Verify changes

**Lab Dependency** – This lab requires the completion of Lab 1.2.

# Lab Topics

- Review course work collected in an Update Set
- Sign up for a developer instance
- Retrieve an Update Set, committing platform changes to the developer instance
- Verify changes

**Lab Dependency** – This lab requires the completion of Lab 1.2.

To end class, you will be walked through the steps to sign up for a developer instance on the ServiceNow Developer Portal.

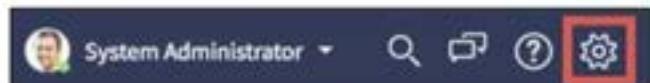
We will also practice the procedure of retrieving completed update sets from one instance of ServiceNow to another. This emulates the experience of taking configuration changes made to the platform of one instance and pulling them into another – much like pulling work from a development instance into production.

## Lab 5.3 Development

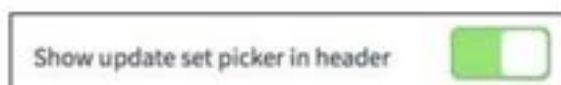
### Review Update Set

You may remember, we created an Update Set in Lab 1.2 which has been collecting updates we have made to the lab instance. We will review some of the updates captured.

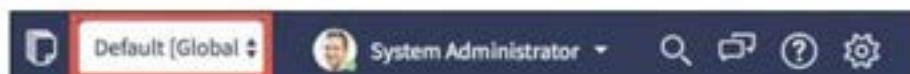
1. Open **Settings** from the banner frame:



2. Next, select **Developer**, then toggle on the switch for **Show update set picker in header**:



3. Close out of the System Settings window and return to your main instance screen. You will now see a drop down menu in the banner frame, next to the user menu:



This is the **Update Set Picker** menu. It allows you to quickly select an update set to capture platform configuration changes.

**NOTE:** If it is not displaying the **Default Update Set**, it should be displaying the **ServiceNow Fundamentals Update Set**.

4. **System Update Sets > Local Update Sets.**
5. Locate and open the **ServiceNow Fundamentals Update Set**.
6. Notice how many total **Customer Updates** have been collected.

Share with the class how many you have in your Update Set.

7. Find the updates for all of the script types created in a previous lab: **Business Rule**, **Client Script**, **UI Policy**, and **UI Policy Action**.
8. Locate the update for the user **Hal Stone**.

What other items are not captured in the Update Set, but that you created during class?

## Register for the ServiceNow Developer Program

Even if you are not a developer or plan to create custom applications in ServiceNow, once you have access to this instance you will be able to use it to revisit the topics discussed here, as well as have fun within *your own* instance of ServiceNow!

1. Go to <https://developer.servicenow.com/>
2. Select **Register** from the upper right-hand menu, then fill out the form, then finally read and agree to terms of use before clicking **Submit**.
3. Look for and open an email from ServiceNow ([sigonon@service-now.com](mailto:sigonon@service-now.com)).
4. Select the link in the email message to validate and activate your account.
5. Sign in with your username and password created in step 3, then read and accept the **ServiceNow Developer Agreement**.
6. Answer a few questions to maximize your experience, then click **Submit**.

7. Under My Instance, click Request Instance:

The screenshot shows the ServiceNow developer program website. At the top, there's a navigation bar with links for PROGRAM, LEARN, API, MANAGE, COMMUNITY, EVENTS, and BLOG. The main header says "Events" and "ServiceNow Developer Program Events". Below the header is a "VIEW EVENTS" button. The main content area is divided into two sections: "My Learning" and "My Instance". The "My Learning" section has tabs for "In Progress" (which is selected) and "Finished". It includes sections for "Learning Plans" (with a message: "You haven't started any Learning Plans. Start one now!") and "Courses". The "My Instance" section contains a message about requesting a ServiceNow instance and a "Request Instance" button. A "JAKARTA" button is also visible.

**NOTE:** If this page does not display, you can click on the ServiceNow logo or, from the main menu, select **Manage > Instance**.

8. Complete the form to tell ServiceNow how you will use the personal developer instance, then click **I understand**.
9. You may choose any available version of ServiceNow you would like, but **Jakarta** is recommended:

A modal dialog box is displayed, asking "Which version of ServiceNow would you like?". It shows three options: "Helsinki" (with a large letter H over a cityscape), "Istanbul" (with a large letter I over a cityscape), and "Jakarta" (with a large letter J over a cityscape). Each option has a "Release Notes" link below it. At the bottom of the dialog, a note says: "If you aren't sure which version to choose, start with Jakarta. It's our latest available release and we have an instance ready for you to get started."

10. After selecting the ServiceNow version of your choice, the credentials to your personal developer instance as well as the URL to the instance will appear on the **My Instance** page:

The screenshot shows the 'My Instance' page for a developer instance named 'dev27768'. It displays the instance status as 'Online', the URL as 'https://dev27768.servicenow.com/', the build as 'Jakarta', and remaining inactivity as '10 days'. Below this are three buttons: 'ACTION +', 'Refresh status', and 'Extend instance'. Underneath these buttons are links for 'Personal developer instance FAQ', 'Report a problem with your instance', and 'Request help from the community'. At the bottom, a green box contains the log-in credentials: 'Username: admin' and 'Password: n4KQ3E9e62yJ'.

**IMPORTANT:**

Capture your username and password as it will not be available after the next step.

11. Under the instance log in credentials, click the link to log into the instance directly:

You can also use this [link](#) to log into the instance directly.

12. Change the temporary passed (captured above) to successfully sign into your instance as system administrator.

## Create an Update Set

Now that you have your course lab instance and a developer instance, practice the method of using Update Sets to move configuration changes between ServiceNow instances.

1. Logged into your course lab instance as System Administrator, navigate to **System Update Sets > Local Update Sets**.
2. Click **New** to open the **Update Sets** form.
3. Enter the following name for the new Update Set: **Problem Form Changes**.
4. In the Description field, enter the following description of this Update Set: **Added Updated and Updated by after Assigned to on Problem form**.

5. Click the **Submit and Make Current** button from the form header.
6. A brief confirmation message displays and the current Update Set is shown:

Your current update set has been changed to Problem Form Changes [Global]

7. This is also represented in the update set picker:



### Modify Problem Form

1. **Problem > Create New.**
2. From the **Form Control Menu** in the Problem header, select **Configure > Form Layout** and make the following changes:  
  
Add **Updated after Assigned to**  
Add **Updated by after Updated**
3. Click **Save**.

### Mark Update Set Complete

1. Click the **View current Update Set** icon, next to the update set picker, to open the update set record:  
  
A screenshot of a user interface showing an update set picker. The 'View current Update Set' icon, which is a red square with a white 'P' icon, is highlighted with a red rectangular overlay. The rest of the picker is white with black text and icons.
2. Notice on the **Customer Updates** tab that a new record is captured.
3. Change the **Update Set State** from **In progress** to **Complete**
4. Click **Update**.
5. With the status of **Complete**, this Update Set is now ready to be retrieved by your developer instance.

## Retrieve Completed Update Sets

1. Return to your developer instance – you may need to log back in as the system administrator.
2. **Problem > Create New.**
3. Observe that the **Updated** and **Updated by** form fields are not displayed on your problem form, in your instance yet:

The screenshot shows the 'Problem' form in ServiceNow. It includes fields for Number (PR0000001), State (Open), Impact (3-Low), Severity (3-Low), and Assignment group (S-Planning). Other visible fields include Business service, Configuration item, Change request, Major problem, Acknowledged, and Short description. There are also 'New' and 'Save' buttons at the top right.

4. **System Update Sets > Update Sources.**
5. Click **New**.
6. Name: **Imported Problem Form Edits**
7. Click the padlock to the right of the URL field.
8. Input your course lab instance URL: <https://instance-####.lab.service-now.com/>
9. Enter this instance's admin **username** and **password**.
10. Type in a brief description of the Update Set into the Short Description field:  
**Modified Problem form fields, adding Updated and Updated by.**

**NOTE:** These are the system administrator credentials your instructor provided, and which you have been using during class.

10. Type in a brief description of the Update Set into the Short Description field:  
**Modified Problem form fields, adding Updated and Updated by.**

11. Your form should look like this:

12. Click the **Test Connection** button.

**NOTE:** A brief **Connection is OK** message will display, which may also include language about varying version snapshots.

13. Save.

14. Click the **Retrieve Completed Update Sets** button in the form header.

15. Close the **Retrieve Update Sets** progress pop-up once it has reached 100%.

### Locate Retrieved Update Sets

1. From the **Retrieved Update Sets** tab, open the **Problem Form Changes** record:

Name	State	Description
Unfiled Record	Previewed	
<b>Problem Form Changes</b>	Previewed	Added Updated and Updated by after Assign...

2. From the form header, click the **Commit Update Set** button.

**NOTE:** This update set has already been previewed – checked to make sure its contents do not interfere with the target instance. If your developer instance is not running Jakarta, you will need to click **Preview** and will get a collision error. You may accept the remote update and **Commit** to continue with the lab.

3. Close the **Update Set Commit** pop-up when its succeeded.

4. **System Update Sets > Retrieved Update Sets.**

5. You should see the new Update Set with a **Committed** state:

ID	Name	Application	Date	Update source	Description	Loaded	Committed
1	Locate/Reset Designer	Global	Previously	Imported Problem Form Edits	2011-07-21 09:47:49	(empty)	
2	Problem form changes	Global	Committed	Imported Problem Form Edits	Added, updated, and updated by after being...	2011-07-21 09:47:49	2011-07-21 09:52:18

### Verify Problem Form Changes

1. **Problem > Create New.**
2. You should see **Updated** and **Updated by** form fields:

The screenshot shows a portion of a form with several input fields. The 'Updated' field is highlighted with a red border, indicating it is the focus of the verification step.

*Congratulations, you have completed the Development Lab!*

## Module Recap: Scripting and Development

servicenow

**Why** would you use these capabilities?

**When** would you use these capabilities?

**How often** would you use these capabilities?

Scripting	Performance and Upgrades	Development
UI/Data Policies	Transaction Log	Update Sets
Client Scripts	Release Cycle	ServiceNow Studio
Business Rules	Upgrades	Integrations
Plugins	Cloning an Instance	