



ServiceNow

# **ServiceNow**

## **Fundamentals**

participant guide



© COPYRIGHT 2019 SERVICENOW, INC. ALL RIGHTS RESERVED.

ServiceNow provides this document and the information therein “as is” and ServiceNow assumes no responsibility for any inaccuracies. ServiceNow hereby disclaims all warranties, whether written or oral, express or implied by law or otherwise, including without limitation, any warranties of merchantability, accuracy, title, non-infringement or fitness for any particular purpose.

In no event will ServiceNow be liable for lost profits (whether direct or indirect), for incidental, consequential, punitive, special or exemplary damages (including damage to business, reputation or goodwill), or indirect damages of any type however caused even if ServiceNow has been advised of such damages in advance or if such damages were foreseeable.

## **TRADEMARKS**

ServiceNow and the ServiceNow logo are registered trademarks of ServiceNow, Inc. in the United States and certain other jurisdictions. ServiceNow also uses numerous other trademarks to identify its goods and services worldwide. All other marks used herein are the trademarks of their respective owners and no ownership in such marks is claimed by ServiceNow.



# ServiceNow Fundamentals

---

## Table of Contents

<b>Module 1: User Interface and Navigation .....</b>	<b>11</b>
Lab 1.1: Applications and Modules .....	27
Lab 1.2: Lists and Filters .....	50
Lab 1.3: Form Configuration .....	74
Lab 1.4: Branding the Instance .....	90
<b>Module 2: Collaboration .....</b>	<b>97</b>
Lab 2.1: Task Management .....	110
Lab 2.2: Notifications .....	126
Lab 2.3: Reporting .....	145
<b>Module 3: Database Administration .....</b>	<b>153</b>
Lab 3.1: Create an Extended Table .....	164
Lab 3.2: Application and Access Controls .....	183
Lab 3.3: Import Data from Excel .....	200
Lab 3.4: Configuration Management Database (CMDB) .....	220
<b>Module 4: Self-Service and Automation .....</b>	<b>229</b>
Lab 4.1: Knowledge Management .....	237
Lab 4.2: Create a Catalog Item .....	256
Lab 4.3: Create a Flow Designer Flow .....	272
<b>Module 5: Introduction to Development .....</b>	<b>285</b>
Lab 5.1: Script a UI Policy and a Business Rule .....	295
Lab 5.2: System Update Sets .....	307
Lab 5.3: Update Source – Update Sets .....	320
<b>Module 6: Capstone Project .....</b>	<b>329</b>
Lab 6.1: Capstone Project Challenge Format .....	334



# ServiceNow Fundamentals

## *The ServiceNow Story*

# What is ServiceNow?

*"What is ServiceNow? It's a platform; a piece of technology that lets people automate workflow in a business."*

**At the heart of ServiceNow lies Fred's vision:  
to make the world of work,  
work better for people.**

*"There is no better experience than giving someone a piece of technology that lets them do something they never thought they could do."*

*– Fred Luddy, ServiceNow Founder*



Who better to explain what ServiceNow is than its founder, Fred Luddy.

*"What is ServiceNow? It's a platform; a piece of technology that lets people automate workflow in a business. It was infuriating to me that IT people could make regular business people feel so silly or ignorant. So for the first 9 months, we built a platform, showed it to a number of people and it was received with a resounding, yawn. Now, it's almost 15 years on, and they're using the platform exactly as we had envisioned."*

*"An entrepreneur is somebody who cares so deeply about solving a problem. I will guarantee you that pretty much everybody that has been one of the tech-types over the past decade and a half was more interested in solving the problem than they were in making money. That's entrepreneurial; doing something which is disruptive and is going to surprise the world in a very wonderful way."*

*"Everything I've ever done in technology has been inspired by something I've seen somebody doing, struggling to do, wishing they could do differently."*

*"Today, ServiceNow's customers include 850 of the Global 2000 biggest companies."* - Forbes Magazine, May 29, 2018



You're in good company!

*“More than 500 companies spend at least \$1 million annually on ServiceNow’s products.” – Forbes Magazine*

Why is ServiceNow so popular? Two things: Simplicity and Customizability.

In this course, you will learn the fundamentals of ServiceNow. You will interact with the versatile Now Platform® and experience how easy it is to configure ServiceNow to match the needs of your organization. This course is just the beginning; the first step in your ServiceNow journey.

## Now Platform®

now.

The Now Platform® provides an Application Platform-as-a-Service (aPaaS), a cloud-based computing model that provides the infrastructure needed to develop, run, and manage applications.

It is not limited to a specific department or function but encompasses the entire enterprise.



ServiceNow's single data model integrates easily with other enterprise systems and supports a wide variety of plug-and-play applications.  
Choose from three workflows or build any workflow app.

A leader in Enterprise Service Management (ESM), 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 (workflows), 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 it can be leveraged to build custom applications.

# Module 1: User Interface & Navigation

now.

## User Interface and Navigation

Collaboration

Database Administration

Self-Service and Automation

Introduction to Development

Capstone Project

## Module Objectives

- Demonstrate how to navigate to applications and modules in ServiceNow using the Application and Filter Navigators
- Create views and filters for a table list
- Update records using inline editing
- Configure a form view using Form Layout and Form Designer
- Describe how to create and update records in a form view
- Understand how to modify the colors in an instance for the logged-in user as well as for all users in the instance

## Labs and Activities

- Lab 1.1: Applications and Modules
- Lab 1.2: Lists and Filters
- Lab 1.3: Form Configuration
- Lab 1.4: Branding the Instance

## Section 1.1: ServiceNow Overview

### User Story

---

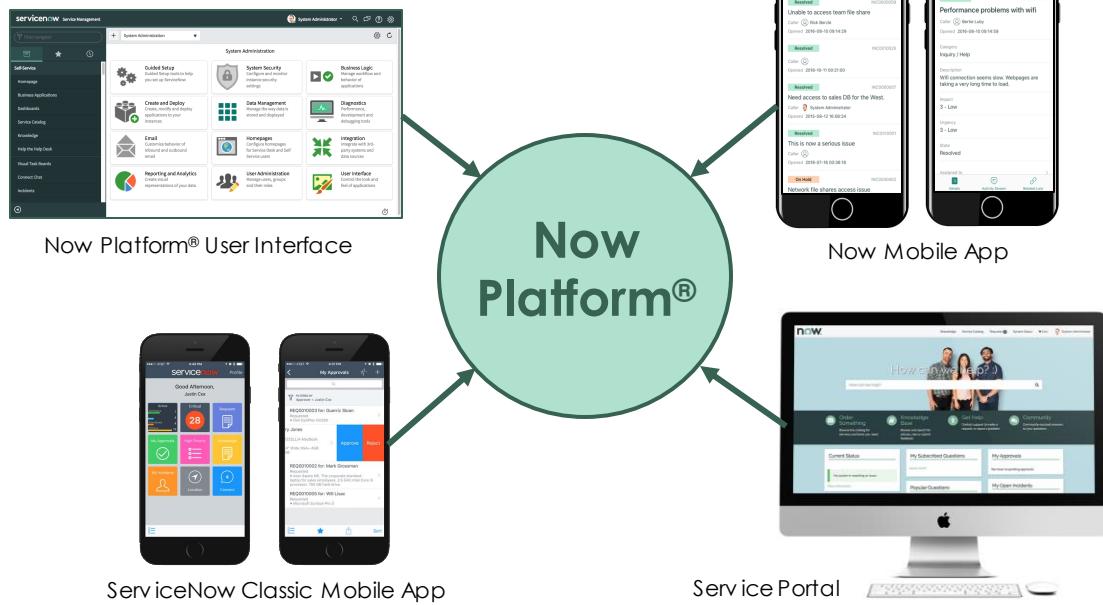
**As users of ServiceNow,**

We want to **become familiar** with the Now Platform interface,

So we can **easily locate applications and modules** and make work, work better for us.



# Now Platform® Interfaces



The **Now Platform® User Interface** is the primary way to interact with the applications and information in a ServiceNow instance.

The **ServiceNow Classic Mobile App** (currently available in the Google Play Store and iTunes) provides the user with the ability to:

- Respond quickly to requests, no matter where they are
- Stay informed with real-time updates and information
- Access the Service Catalog
- Use Connect Chat to collaborate with others

The **Service Portal** provides a user-friendly self-service experience by providing access to specific features using widgets. When accessing the portal via web browser, <https://<instancename>.service-now.com/sp> users can:

- Search for articles, catalog items, records
- Submit requests
- Browse the corporate news feed, and more!

**Now Mobile App** enables employees to submit incidents and requests, manage tasks, and access company resources from anywhere. Through the Now Mobile App, a user can also:

- Upload images and attachments
- Sign documents
- Take surveys

## What is the ServiceNow Instance?

When you are accessing ServiceNow, you are accessing an **instance** of the platform. An organization can have several instances (e.g. Dev, QA, Test, Prod) and each instance is a single implementation of the ServiceNow platform.

What it is...	Why it's cool...
Independent, changeable, and highly configurable	If you can dream it, you can do it!
Not shared with other ServiceNow customers (single-tenant)	Your data isn't mingled in the same database as another organization's data.
Each instance has applications	You choose which products you want to purchase so you don't have the overhead of the entire suite.
Each instance has customer data that can be exchanged between instances	It's easy to migrate data from dev to QA to test to stage to production.
Upgrades are made on individual instances	Upgrade a non-production instance to complete testing before upgrading production.

A load-balanced instance is located (hosted) in one of the ServiceNow Data Centers around the world, or for a very, very small percentage of our customers, an instance can be implemented onsite at the customer's location. Each ServiceNow instance has a unique URL that uses a format similar to <https://<instance name>.service-now.com>

ServiceNow utilizes an advanced, multi-instance, single-tenant architecture as the default offering for customers, meaning an instance features an individually isolated database containing data, applications, and customizations.

The ServiceNow multi-instance architecture, organized in an instance stack, 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.

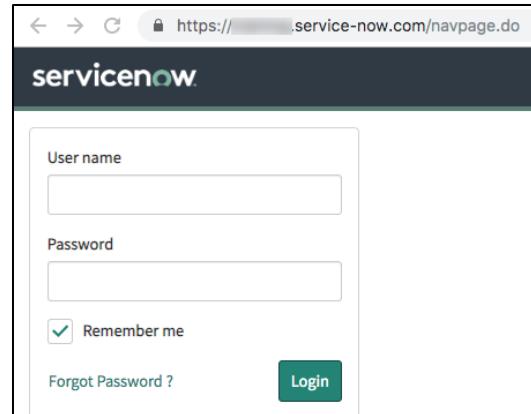
Each customer organization receives two instances of ServiceNow: production and sub-production. They have the ability to obtain additional sub-production instances to be used for User Acceptance Testing (UAT), Review, Development, or Quality Assurance (QA).

# Logging in to the Now Platform User Interface®

now.

1. Navigate to instance URL:  
`https://<instance_name>.service-now.com`
2. Enter **User name**
3. Enter **Password**
4. (Optional) Check **Remember me**
5. Select **Login**

User authentication is the first level of security applied for accessing data in ServiceNow.



A screenshot of a web browser displaying the ServiceNow login page. The URL in the address bar is `https://<instance_name>.service-now.com/navpage.do`. The page has a dark header with the "servicenow" logo. Below the header is a form with two input fields: "User name" and "Password". Underneath the password field is a checked checkbox labeled "Remember me". At the bottom left of the form is a link "Forgot Password ?" and at the bottom right is a green "Login" button.

To access the platform interface, a user navigates to a secure, unique URL: `https://<instance_name>.service-now.com` from a supported web browser. For information on supported browsers, visit the ServiceNow documentation website, <https://docs.servicenow.com> and search for Browser support. The ability to use a custom URL is not supported. However, an administrator can associate a custom URL that resolves to the instance name.

Users are authenticated by various methods, including:

- **Local database:** The user name and password in their user record in the instance database
- **Multi-provider Single Sign-on:** Allows the selection/use of several identity providers (IdPs) to manage authentication as well as retain local database authentication.
- **Multifactor:** The user must enter a passcode or token in addition to the password. A mobile application, Google Authenticator, on a user mobile device generates the passcode.
- **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

# Users and Groups

## Users

Within a ServiceNow instance, **users** are:

- Updating records
- Importing data
- Requesting items
- Implementing flows
- Approving knowledge content
- Running reports
- Developing applications



Users are represented by a record on the  
**User [sys\_user]** table

## Groups

A collection of users is a **group**

Groups share a common purpose such as users approving change requests or users receiving e-mail notifications

Examples of Groups include:

- Service Desk
- Knowledge Base Authors
- HR Administrators



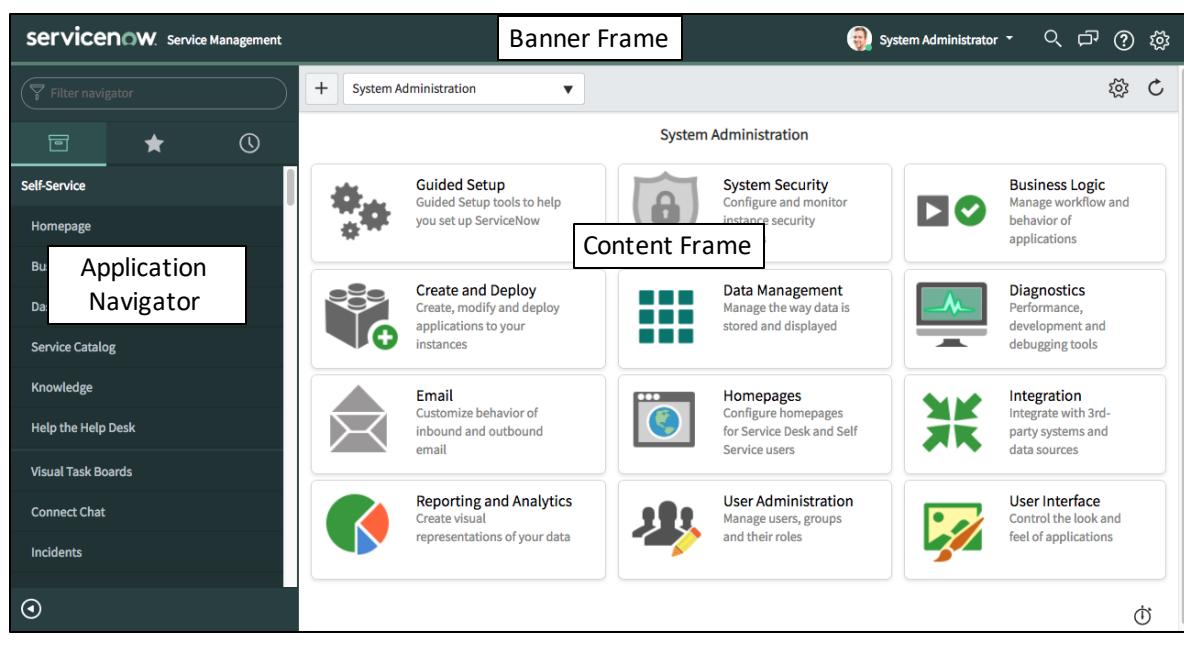
A group is represented by a record on the  
**Group [sys\_user\_group]** table

Manage the individuals who can access ServiceNow by defining them as users in the platform. User IDs are unique identifiers for the user's ServiceNow login user name. 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.

Later in the course, you will learn how roles are used in conjunction with users and groups as the second level of security for accessing data in ServiceNow.

# Now Platform® User Interface

now



The User Interface (UI) is the main way for users 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.

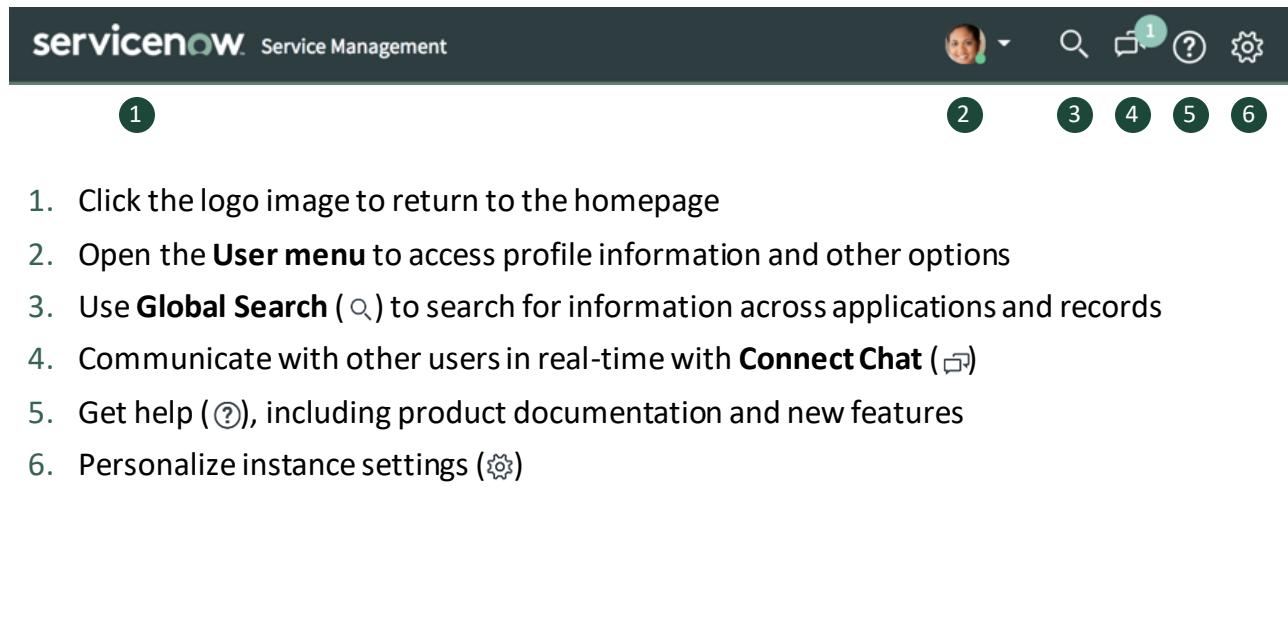
The ServiceNow user interface is divided into three areas:

- 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.

## Banner Frame

now.



Across the top of the platform interface, the *banner frame* is displayed. The banner frame contains the following components:

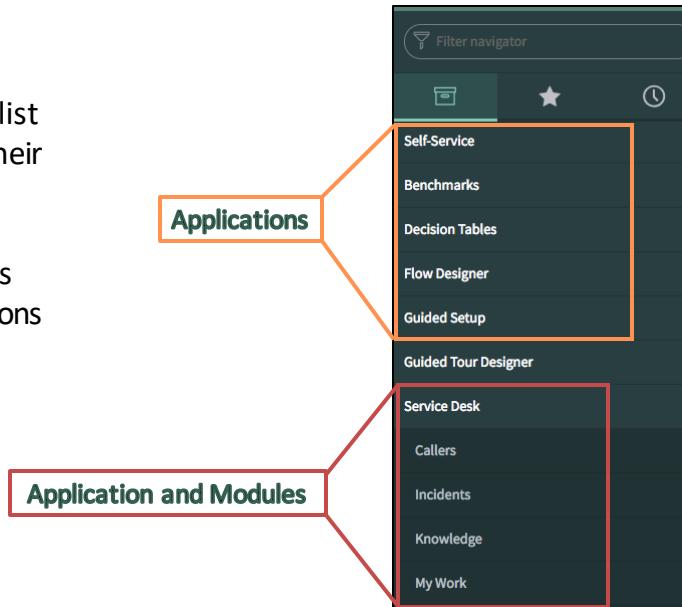
1. **Logo image** (can be modified for the instance) returns the user to the homepage when clicked.
2. **User menu** provides options to access your profile and preferences. Administrators can impersonate users and elevate their security role from the User menu.
3. **Global text search icon**: Finds records from multiple tables (quickest way to locate an individual record)
4. **Connect sidebar icon**: Lets you begin or continue conversations. This icon is visible if Connect is enabled.
5. **Help icon**: Opens the help panel with embedded help, where available. If there is no embedded help, it offers help search options.
6. **Gear icon**: Opens the **system settings for the user interface (UI)**. For more information on system settings, search ServiceNow's Product documentation ([docs.servicenow.com](https://docs.servicenow.com)) for the article titled **System settings for the user interface (UI)**. NOTE: Changes made to the system settings using the Gear icon are applicable only to the logged-in user.

# Application Navigator

The **Application navigator** is a list of available applications and their corresponding modules.

The following four components can be used to locate applications and modules:

- Filter navigator
- All applications tab
- Favorites tab
- Your history tab



The application navigator, or left navigation bar, provides access to all applications and the modules they contain. An application is a group of modules, or pages, that provide related information and functionality in an instance. For example, the Incident application contains modules for creating and viewing incidents. The Configuration Management application contains modules for configuring servers, databases, and networks.

There are four components in the application navigator which help a user quickly find information and services:

- Filter navigator
- All applications
- Favorites
- Your history

**Tip:** Double-click the **All applications** icon to expand or collapse all applications in the navigator

## Application Navigator: Filter Navigator

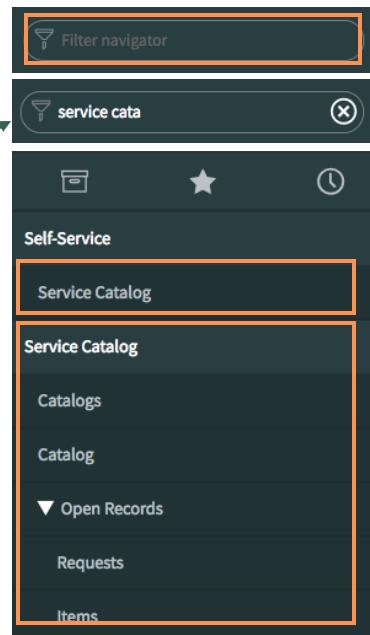
Above the application list, use the **Filter navigator** to quickly navigate to applications and modules

Simply begin typing the application or module name\*

View the results:

- All modules with the keyword display
- All modules within an application that contains the keyword display

\*All, or part, of any module name



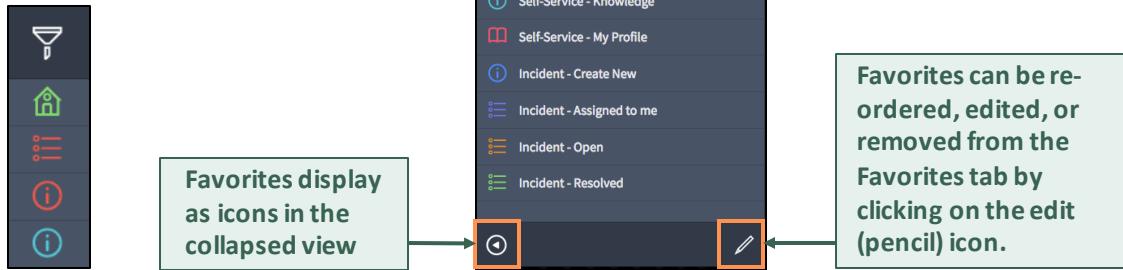
Using the Filter navigator, a user can:

- Quickly navigate to applications or modules by typing all, or part, of the application or module name (e.g. Typing **ge ba** in the filter navigator displays the Knowledge Bases module)
- Navigate directly to a table using:
  - `<table name>.list` – displays a list of records in the table
  - `<table name>.form` – displays a blank form for creating a new record in the table

**Just for fun!** Try **kb\_knowledge.CONFIG** in the filter navigator. Notice a separate tab opens in your browser window. What kind of information is shown for the table?

# Application Navigator: Favorites tab

The **Favorites** tab, represented by a star icon, displays items you have added as favorites.



Favorites can be added when you:

- Click the star icon by a module
- Click the star icon by an application
- Select **Create Favorite** from a list or form context menu
- Drag a breadcrumb to the **Favorites** tab of the application navigator

Every user can collapse and expand the application navigator by clicking on the arrow icon at the bottom left of the application navigator. When the application navigator is collapsed, the filter (funnel) icon and the favorites icons are displayed at the left of the user interface. To expand the application navigator, click the filter icon or the arrow icon.

More information for adding/editing favorites can be found by searching ServiceNow's product documentation ([docs.servicenow.com](https://docs.servicenow.com)) for the article titled **Use the UI16 navigator**.

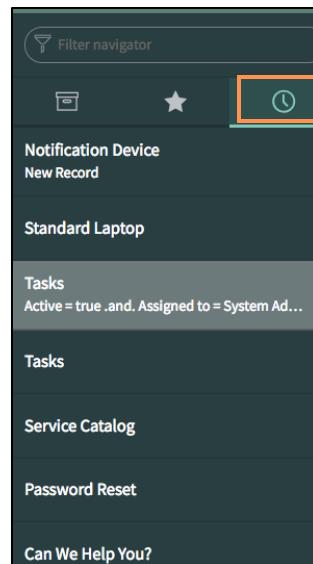
**Tip:** Admins can view favorites set by any user by navigating to **System Definition > Bookmarks**.

## Application Navigator: Your History

**Your History**, represented by a clock icon, is a scrolling view of recent activities including:

- Forms and lists accessed
- Catalog items
- Homepages visited

Click on any recent activity to open the item in the Content Frame.



By default, the maximum number of activities displayed in the history tab of the application navigator is 30. Administrators can configure this setting in the system properties table (`sys_properties.list`) by modifying the value in the `glide.ui.nav.history_length` property.

The system creates history entries for many types of content, including lists, records, and homepages. Some content types are not tracked in the history, such as UI pages and other non-standard interfaces.

# Content Frame: Common Types of Interfaces

now.

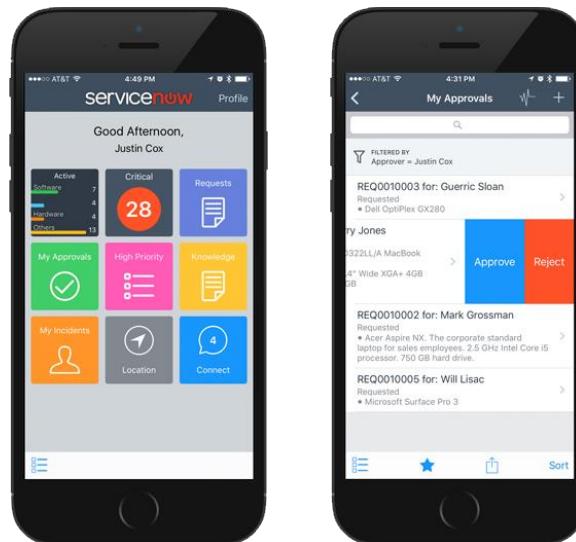
The image displays six ServiceNow interface examples arranged in a 2x3 grid:

- Homepage:** Shows a navigation menu with categories like System Administration, Business Logic, Data Management, and Reporting and Analytics.
- List:** Displays a list of incidents with columns for ID, Title, Status, Priority, and Due Date.
- Form:** Shows a detailed view of an incident record with various fields and tabs.
- Dashboard:** Features three main metrics: Open Risk (256), Open Issues (125), and Active Projects (61), along with bar charts for each category.
- Knowledge Base:** Lists knowledge bases such as Instance Security Center, IT, Knowledge, and Social QA.
- Service Catalog:** Shows a catalog of services including Hardware, Software, Office, Desktops, Peripherals, and Mobiles.

# ServiceNow Classic Mobile App

now.

- Respond quickly to requests, no matter where you are
- Stay informed with real-time updates and information
- Save favorite lists and records to the app home screen
- Access the Service Catalog
- Use Connect Chat to collaborate with others
- Share your location



**Mobile Favorites** – Favorites display as icons on the homepage and provide links to records in the system. Favorites automatically synchronize between the mobile UI and the desktop UI. Long press a favorite to edit the following properties:

- Label
- Text and icon color
- Icon image

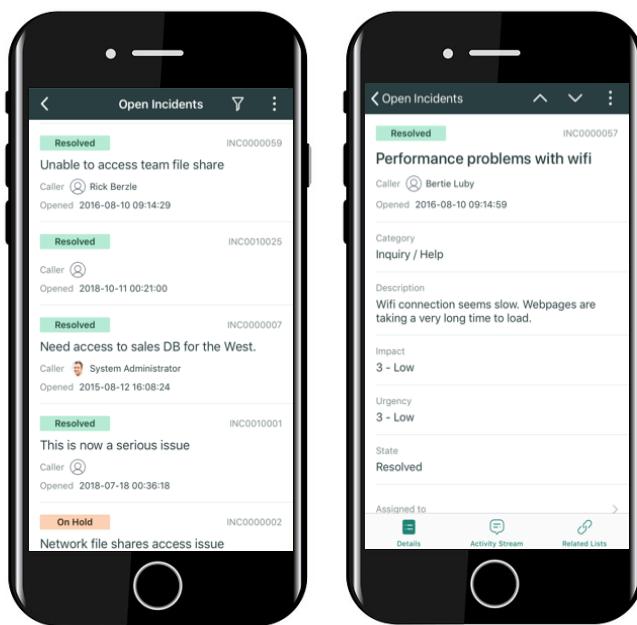
**Mobile Lists** – Lists appear as a single column of records on the mobile app UI display. Each row represents a separate record. Live list updates are not available on mobile devices when other users make changes. Mobile lists automatically reload when a record change is made by the currently logged in user.

**Mobile Service Catalog** – Browse the service catalog using a browser on your mobile device or with the ServiceNow Classic mobile app. The mobile UI for Service Catalog contains the following elements:

- Current catalog
- Back button
- Search
- Application navigator button
- Shopping cart

**Mobile Connect Chat** – Communicate with coworkers using the ServiceNow Classic mobile app. Many features supported in the desktop interface are available in the mobile app.

## Now Mobile App



The Now Mobile app is a mobile-first, fully native experience where mobile users can:

- Submit and view requests and issues
- Submit approvals
- View and complete tasks
- Upload images and attachments
- Sign documents
- Request help from Agent Chat

Documentation regarding Now Mobile App can be found by searching ServiceNow's product documentation ([docs.servicenow.com](https://docs.servicenow.com)) for the article titled **Use the Now Mobile app**. From there you can learn how to:

- Enable notifications
- View and complete tasks
- Submit and track requests
- Search for articles, people, and services
- Report an issue
- Ask for help

# Product Documentation: Docs and Community

now.

The screenshot shows two main sections. On the left is the 'Now Platform Capabilities' page under the 'Contents' menu. It features a video player showing a 'Getting Started | Platform Architecture' video titled 'ServiceNow's System of Action'. The video thumbnail displays various ServiceNow modules like IT, Security, Customer Service, HR, and Intelligent Apps. Below the video, there are sections for 'Cloud Services', 'Now Platform', and 'Nonstop Cloud'. On the right is the 'Community' website interface, which includes a search bar, navigation links for Communities, Groups, Resources, Events, Blogs, and Knowledgebase, and a 'Welcome to the Community' message. Below the header, there are sections for 'Browse Community Forums' and various management categories.

If anything in this class seems interesting, we highly encourage you to explore the topic in more detail through either of the following websites:

**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** 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!



**Time**  
**10-15m**

- Log in to your training instance
- Use the Application Navigator and its filter to access different areas of ServiceNow
- Add Knowledge and Service Catalog modules to Favorites

## Lab 1.1: Applications and Modules

As **users of ServiceNow**,

We want to **become familiar** with the Now Platform interface,

So we can **easily locate applications and modules** and make work, work better for us.

# Applications and Modules

Lab  
1.1

10-15m

## Lab Objectives

You will achieve the following objectives:

- Log on to your training instance
- Use the Application Navigator and its filter to access different areas of ServiceNow
- Add Knowledge and Service Catalog modules to Favorites

## Scenario

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

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

ServiceNow will initially be used by Cloud Dimensions for tracking Infinity inventory, order fulfillment, and customer support.

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

To start, you will assume the system administrator identity to accomplish a series of tasks.

## A. Log on to Your Student Instance

1. Navigate to **your assigned ServiceNow Lab Instance** in the web browser of your choice. Make note of the instance URL, you will be using it later in the course.

**Note:** Your instructor will provide you with a registration URL for requesting a student instance.

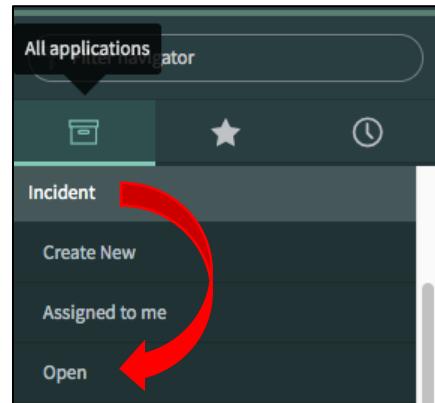
2. Log on using the **System Administrator (admin) credentials** provided when you requested your student instance using the registration URL.

## B. Use the Filter Navigator

1. Take some time to scroll through the Application Navigator and notice the available application menus and modules.
2. 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.
3. Navigate to **Incident > Open**

**Note:** The **Application Menu > Module name** formatting indicates the navigation path to use in the expanded Application Navigator; for this step, navigate to the **Incident** application and select the **Open** module. This shorthand instruction will be used going forward in the class.

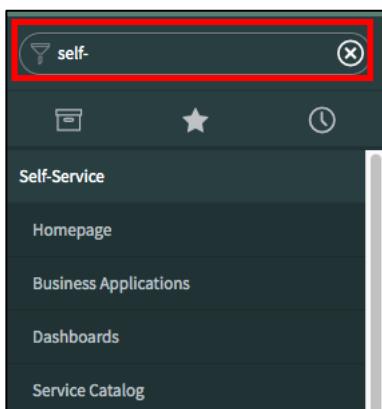
Notice how the information displayed in the Content Frame changed from a homepage to a list.



4. **Incident > Create New**

**Note:** Notice how the information displayed in the **Content Frame** changed from a list of incident records to an individual incident record/form.

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

6. **Self-Service > Service Catalog**

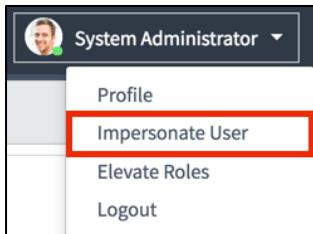
**Note:** Notice yet another user interface type displayed in the **Content Frame**.

7. 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.

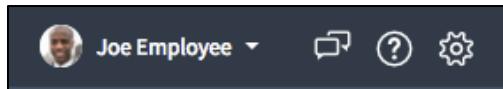
## A. Set Module Favorites

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



2. Impersonate the Cloud Dimensions employee **Joe Employee** 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 Joe Employee:



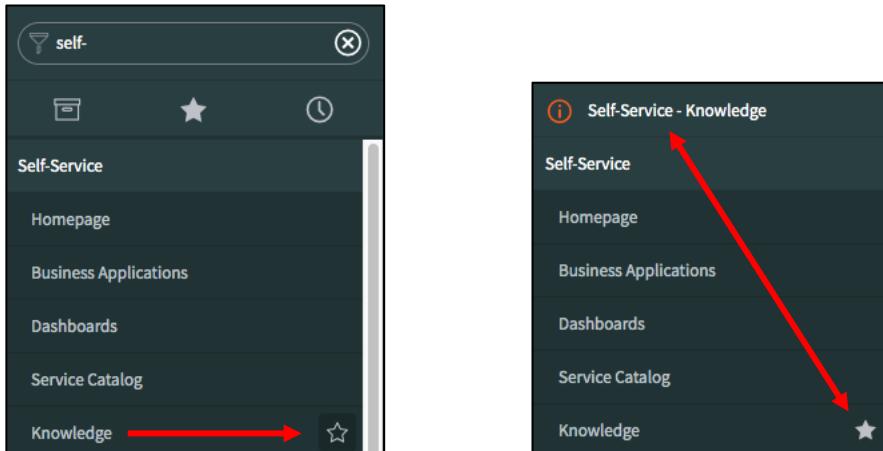
3. Take some time to scroll through the Application navigator and notice the available application menus and modules compared to those seen by the system administrator.
4. Filter the **Application Navigator** using the keyword *self*.
5. **Self-Service > Knowledge**
6. Open the **ServiceNow Fundamentals Class** Knowledge Base.



7. Download the necessary lab files for class by selecting the **Course Files** article.

**Note:** The Course Files article contains a hyperlink to access a folder containing two zipped files: **Capstone Files.zip** and **Class Lab Files.zip**. At your convenience, download both files and unzip the Class Lab Files. The Capstone Files will be used in Module 6 of this course.

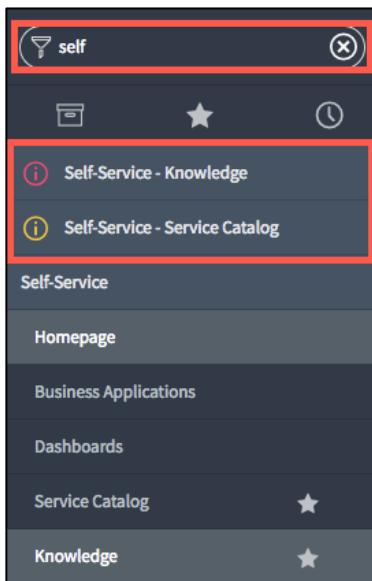
8. Hover over the **Knowledge** module, then add the **Knowledge** module as a favorite by selecting the **Add to Favorites** icon (star) to the right of **Knowledge**.



**Note:** Upon selection, the **Add To Favorites** star will appear filled in and the favorite will display at the top of the Application Navigator.

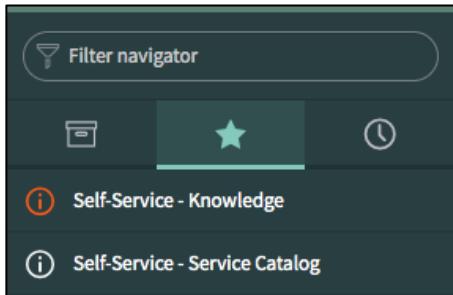
9. Repeat this step for the **Service Catalog** module.

**Note:** In addition to application menus and modules, the **Filter navigator** will display favorites based on keywords:

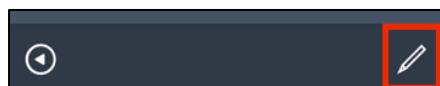


10. Clear the Filter navigator keyword by selecting the X to the right of the Filter navigator.

11. Next, navigate to the **Favorites** tab of the Application Navigator to see the module favorites you have created:



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



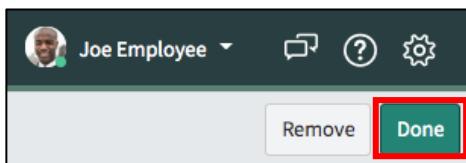
**Note:** The *Set up your favorites* screen displays in the Content Frame. A favorite can be customized to have any name, color, and icon.

13. Select the **Self-Service - Knowledge** favorite in the **Application navigator**.

14. Select any color and icon for the **Self-Service > Knowledge** favorite.

15. Repeat steps 12 and 13 to add **Self-Service > Service Catalog** as a favorite.

16. Click the **Done** button:



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

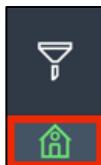


18. From the minimized Application Navigator, notice that the two favorites appear in the color and icon you have selected:

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



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



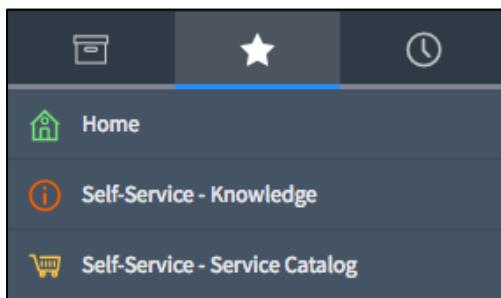
20. Select **Maximize Navigator** at the bottom of the minimized Navigator:



## Lab Verification

**Note:** The Lab Verification section displays one, or more, screenshots of what you should have created during this lab. Sometimes the lab verifications have been shown in earlier steps – as is the case below.

### Self-Service Favorites



**Congratulations on completing the Applications and Modules lab!**

**Now you know how to create and maintain application and module favorites!**

## Knowledge Check

What are the different interfaces for accessing a ServiceNow instance?

- Now Platform® User Interface
- Service Portal
- ServiceNow Mobile Classic
- ServiceNow Agent

What are the three components of the Now Platform® User Interface?

- Banner Frame
- Application Navigator
- Content Frame



What are the components of the Application Navigator?

- All applications
- Favorites
- Your history
- Filter navigator

What is a collection of users who share a common purpose?

- Group

Where can you locate product documentation?

- [docs.servicenow.com](https://docs.servicenow.com)

## Section 1.2: Lists and Filters

### User Story

---

As the **Service Desk Manager**,  
I need an **Infinity specific list view of incidents**  
So senior resources can **limit time spent finding and updating** Infinity incident records.



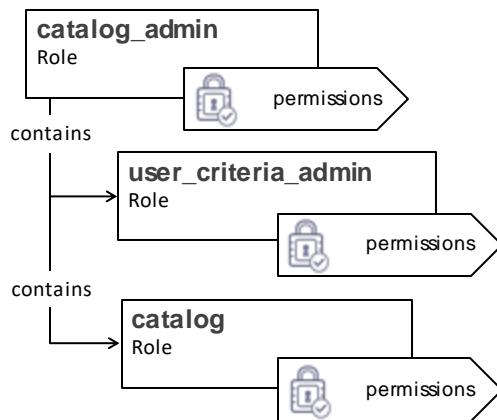
A **role** is a *collection of permissions* used to:

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

It can be assigned to a group\* or a single user; a user can have more than one role.



Roles are represented by a record on the Role [sys\_user\_role] table



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

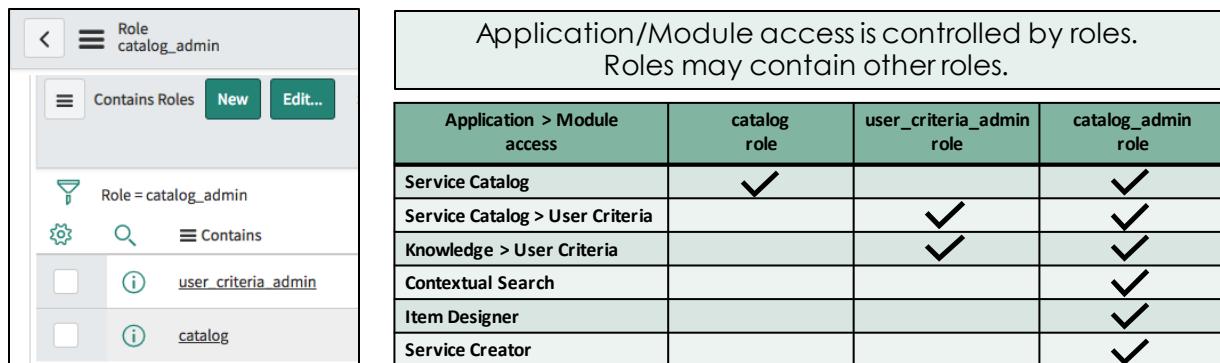
Additionally, a role may contain other roles and any access that is granted to one role is automatically granted to any role that contains it. By assigning a role to a group, all the users in the group inherit all of the roles within that role.

**\*TIP:** Rather than adding roles to individual users, add the user to a group and assign the role to the group. This method of role assignment makes maintenance easier when people transfer to different roles in the organization. A user can be removed from one group and added to a new group; inheriting all the permissions needed to perform their duties.

**NOTE:** You cannot delete roles that are assigned to the group from a user record. You must remove the user from the group record.

## Roles

In this example, we see the **catalog\_admin** role contains two roles: the **catalog** role and the **user\_criteria\_admin** role.



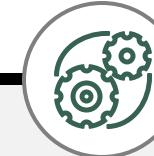
The screenshot shows the ServiceNow Role Catalog interface. On the left, a card for the 'catalog\_admin' role is displayed, showing it contains the 'catalog' and 'user\_criteria\_admin' roles. On the right, a table titled 'Application/Module access is controlled by roles. Roles may contain other roles.' illustrates the access levels for various applications and modules across three roles: catalog role, user\_criteria\_admin role, and catalog\_admin role.

Application > Module access	catalog role	user_criteria_admin role	catalog_admin role
Service Catalog	✓		✓
Service Catalog > User Criteria		✓	✓
Knowledge > User Criteria		✓	✓
Contextual Search			✓
Item Designer			✓
Service Creator			✓

The table illustrates the following:

- A user who is assigned the **catalog** role only has access to the **Service Catalog** application and its associated modules.
- A user with the **user\_criteria\_admin** role only has access to two modules: **Service Catalog > User Criteria** and **Knowledge > User Criteria**.
- A user with the **catalog\_admin** role has access to all of the applications and modules of both roles plus any permissions (access) specific to the **catalog\_admin** role.

## Base System Roles

				
<p><b>System Administrator</b></p> <p>The <b>admin</b> role provides access to all platform features, applications, functions, and data</p>	<p><b>Specialized Administrator</b></p> <p>Users with specialized administrator roles may manage specific functions or applications, including:</p> <ul style="list-style-type: none"> <li>Assignment Rules</li> <li>Knowledge Base</li> <li>Human Resources</li> <li>Reports</li> <li>Web Services</li> </ul>	<p><b>Fulfiller</b></p> <p>Users with the fulfiller (<b>itil</b>) role may fulfil ITIL activities associated with the ITIL workflow, including incident and change management</p>	<p><b>Approver</b></p> <p>The <b>approver_user</b> role can perform all requester actions and allows users to view or modify approval records directed to them</p>	<p><b>Requester</b></p> <p>Also known as Employee Self Service (ESS) users, these users do not have roles but can submit and manage their own requests, access public pages, etc.</p>

ServiceNow baseline includes many base system roles; some of which are shown here. To get more information on base system roles, please search ServiceNow product documentation and search for the title, **Base system roles**.

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.

- **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.

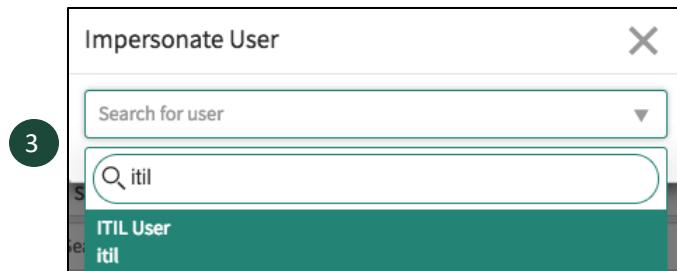
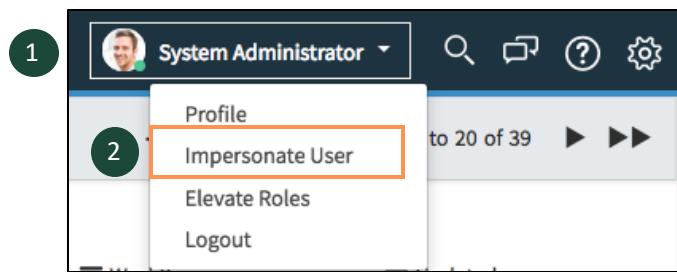
**NOTE:** The **impersonator** role can be assigned to a user to allow impersonation of other users, excluding admins, for testing and visibility purposes.

# Impersonate a User

Users with the **admin** or **impersonator** role can impersonate other users for testing purposes.

To impersonate a user:

1. Open the user menu by clicking your user name
2. Select **Impersonate User**
3. Select a user from the recent impersonations list or enter the user's name to search for the user



When impersonating a user, the impersonator can\*:

- Access exactly what that impersonated user can access (applications, modules, data)
- Test what different users can do in ServiceNow

**NOTE:** Impersonations are logged in the System Log.

\*Not all users can be impersonated. If an admin attempts to impersonate a user with a specialized administrator role for Human Resources or Security Incident Response, they will not be able to access features granted by that role.

**TIP:** It is recommended to create logins for the following roles to effectively test the system:

- admin – to do work
- itil – to test as a fulfills
- ess – to test as an end user

## Viewing Table Data: Lists

A **list** displays a set of records from a table within the content frame of ServiceNow.

The screenshot shows the ServiceNow User Administration application. On the left is a sidebar with various navigation options like User Administration, Role Attributes, Users, Groups, Roles, and more. The main area is titled 'Users' and shows a list of user records. A callout box highlights a specific row with the text 'Each row represents one record'. Another callout box highlights a specific column with the text 'Each column represents one field'.

User ID	Name	Email	Manager	Active	Created	Updated
abel.tuter	Abel Tuter	abel.tuter@example.com	(empty)	true	2012-02-17 19:04:52	2019-08-20 11:46:26
abraham.lincoln	Abraham Lincoln	abraham.lincoln@example.com	(empty)	true	2013-07-23 17:15:54	2019-08-20 11:46:31
adela.cervantes	Adela Cervantes	adela.cervantes@example.com	(empty)	true	2012-02-17 19:04:50	2019-08-20 11:46:19
aileen.mottern	Aileen Mottern	ailene.mottern@example.com	(empty)	true	2012-02-17 19:04:49	2019-08-20 11:46:26
alejandra.prenatt	Alejandra Prenatt	alejandra.prenatt@example.com	(empty)	true	2012-02-17 19:04:52	2019-08-20 11:46:20
alejandro.mascall	Alejandro Mascall	alejandro.mascall@example.com	(empty)	true	2012-02-17 19:04:52	2019-08-20 11:46:30
alene.rabeck	Alene Rabeck	alene.rabeck@example.com	(empty)	true	2012-02-17 19:04:53	2019-08-20 11:46:32
alfonso.grigjen	Alfonso Grigjen	alfonso.grigjen@example.com	(empty)	true	2012-02-17 19:04:51	2019-08-20 11:46:19
alissa.mountjoy	Alissa Mountjoy	alissa.mountjoy@example.com	(empty)	true	2012-02-17 19:04:52	2019-08-20 11:46:26
allan.schwartz	Allan Schwartz	allan.schwartz@example.com	(empty)	true	2012-02-17 19:04:53	2019-08-20 11:46:31
allie.pumphrey	Allie Pumphrey	allie.pumphrey@example.com	(empty)	true	2012-02-17 19:04:52	2019-08-20 11:46:31
allyson.gillispie	Allyson Gillispie	allyson.gillispie@example.com	(empty)	true	2012-02-17 19:04:50	2019-08-20 11:46:18
alva.pennington	Alva Pennington	alva.pennington@example.com	(empty)	true	2012-02-17 19:04:50	2019-08-20 11:46:33

Lists and forms are the most common ways to interact with data. A list displays a set of records from a table. Lists can be filtered and customized to display the information you need.

In this example, the system administrator is accessing the **User Administration** application and a list of users through the **Users** module. Other roles, such as **user\_admin**, grant users the permissions to manage users, groups, and roles.

# List Anatomy

## 1. Title Bar:

Displays the list title and, in some cases, the view name as well as search list values and a record count

## 2. List Filters/Breadcrumbs:

Offers a quick form of filter navigation

## 3. Column Headings:

Displays column (table field) names and provides some list controls

## 4. Column Header Search:

Provides a search within a specific column

## 5. Field Values:

Right-click on a field value to access additional actions

The screenshot shows a list view in ServiceNow with the following features highlighted:

- 1.** Title Bar: Shows the list title "Knowledge" and a "New" button.
- 2.** List Filters/Breadcrumbs: Shows filters like "All" and "Number".
- 3.** Column Headings: Shows column headers such as "Short description", "Author", "Category", "Workflow", and "Updated".
- 4.** Column Header Search: Shows a search bar within the column header area.
- 5.** Field Values: Shows a context menu (Resources) for a field value in the "Category" column.

The list displays 10 records with columns including Number, Short description, Author, Category, Workflow, and Updated.

Number	Short description	Author	Category	Workflow	Updated
KB001110	ServiceNow Secure Coding guide for Insta...	System Administrator	Best Practices		12/31/17 16:00:00
KB001109	Customer application penetration testing	System Administrator	Resources		12/31/17 16:00:00
KB001108	ServiceNow TRUST & Compliance Center	System Administrator	Resources		12/31/17 16:00:00
KB001106	Instance Security Center - Resources	System Administrator	Resources		12/31/17 16:00:00
KB001105	Platform Security (Security Docs)	System Administrator	Resources		12/31/17 16:00:00
KB001104	Instance Hardening Guide	System Administrator	Best Practices		12/31/17 16:00:00
KB000033	Eclipse configuration for Android develo...	System Administrator	Android		06/11/15 10:10:57
KB000032	Getting Around in Windows	System Administrator	Java		06/11/15 10:10:53
KB000031	How can I find the MAC address of my Eth...	Sam Sorkin	How To		12/19/14 07:49:16
KB000030	Deleted Email Recovery	Ron Kettering	Outlook 2010		11/14/18 13:51:24

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

# Context Menus

Context menus provide different levels of controls for a given list view

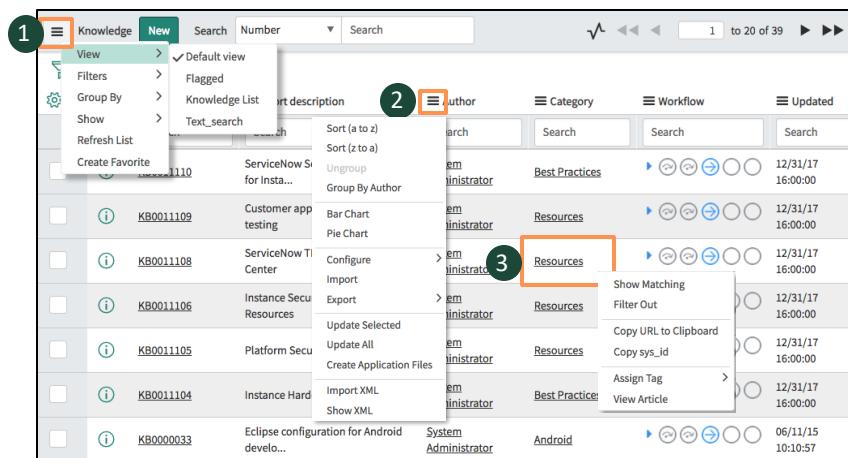
## 1. List Context Menu

## 2. Column Context Menu

## 3. Record Context Menu\*

Context menus can be accessed by clicking the list menu icon or by right-clicking the list header and column headers respectively

\*Right-click in a row's cell



List Context (or control) 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 Context Menu:** Click the list context 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 Context Menu:** Click the column context 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 Context Menu:** 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 Filters

A **filter** is a set of conditions applied to a table list to isolate a subset of the data

Three components that make up a **filter condition** include:

**1. Field**

**2. Operator**

**3. Value**

The screenshot shows the filter configuration interface. At the top, there are buttons for Knowledge, New, Search, Number, and another Search button. Below that is a toolbar with Run, Save..., AND, OR, Add Sort, and a delete icon. The main area displays the filter conditions: "All of these conditions must be met". There are three conditions listed, each with a circled number (1, 2, 3) and a circled delete icon (X). Condition 1: Active is true. Condition 2: Category(kb\_category) is Resources. Condition 3: Author is Ron Kettering.

Click the show / hide filter icon ( to add, remove, or edit **filter conditions** and apply



**Tip:** Add filters to your **Favorites** by dragging the breadcrumb to the navigator

All users can apply, create, modify, and save filters. You may start with a list of all incidents but filter those records to view only active incidents assigned to you.

Select **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 context menu.

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 modify conditions and add to your favorites directly from the breadcrumbs .

The three parts of a filter condition are:

1. **Field:** A choice list based on the table and user access rights. The choice list includes fields on related tables by dot-walking.
2. **Operator:** A choice 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.

Filter operators will change depending on field data type. Example field operators include:

- **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

# List Editing

The **List Editor** allows a field value to be edited in a list without opening the record

Locate a record with the field value 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

Name	Short description	Catalogs	Category
Service Category Request	Start managing your own service requests	Service Catalog	Departmental Services
Access	Microsoft Access	Service Catalog	Software
Standard Laptop	Lenovo - Carbon x1	Service Catalog	1 Hardware
Apple iPad 3	Apple iPad 3	Service Catalog	Tablets

Service Catalog	Software	\$139.99	Item
Service Catalog	2 Laptops	3	Cancel

Service Catalog	Laptops
-----------------	---------

Users can edit data in lists using various methods but certain field types cannot be edited. Additionally, list editing is disabled for some tables.

The list editor is the quickest method to update a field on multiple records.

## Procedure

1. Select the records to be edited.
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 will be edited 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 that are to be tracked based on a user-defined label

# Tags

Use **tags** to categorize, flag, and locate records

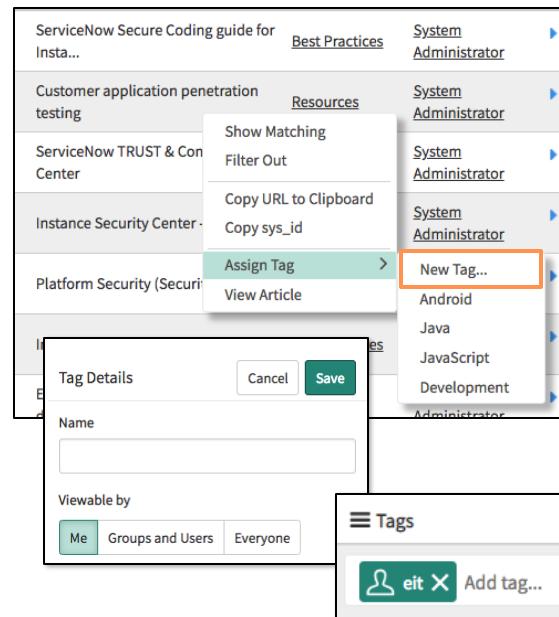
Tags can be created against any record from a list or form view

New tags can be made visible to:

- The current logged in user (Me)
- Groups and Users
- Any user (Everyone)



Edit personal tags by using the **My Tags** or **My Tagged Documents** modules.



There are a few ways to assign tags to records:

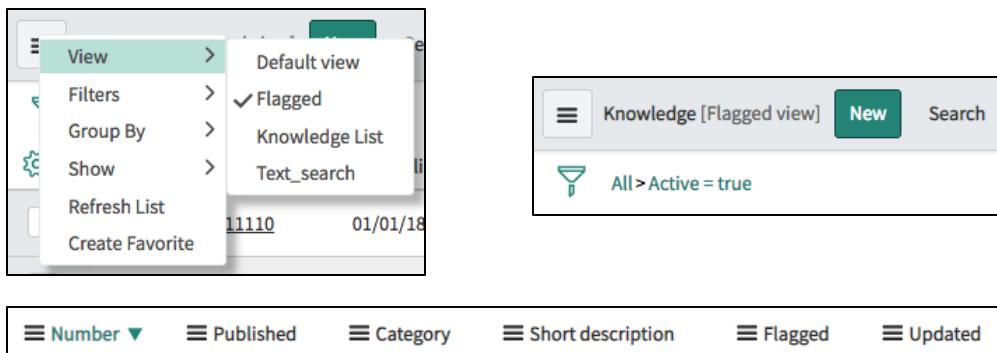
- From the list view using inline field editing
- From a list using the record context menu
- Configuring tags to assign automatically

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, a user must have the **admin** or **tags\_admin** role.

## Views

A **view** is a version of a customized list or form which defines which fields appear and in what order. Different views can be created and used for different roles.



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

**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, different views can be created and used on 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 Context 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. A message displays asking to save or discard all changes made to the record, before the form reloads and displays the selected view.

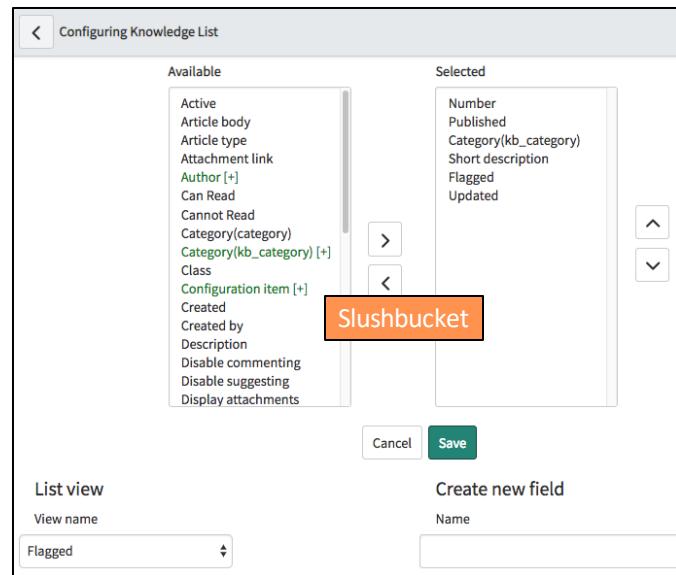
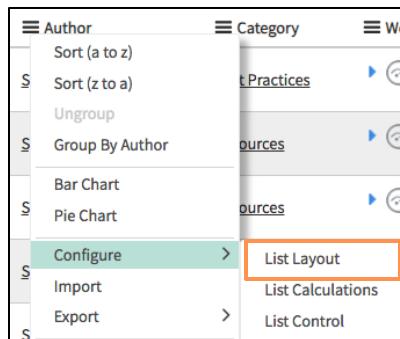
**Sort Controls:** A list that is displayed to a user for the first time will be sorted by one of the following:

- 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

## Views: Layout Configuration (all users)

Configure the **List Layout** to show or hide fields from a view and change the list column order

Click any column context menu icon and select **Configure > List Layout**



Users with the **admin** or **personalize\_list** role 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 configure the List Layout for a table:

1. Navigate to the list
2. Click any **column context menu**
3. Select **Configure > 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 box, you can adjust their order (up or down) on the list.

**NOTE:** To create a new list view, select **New...** from the **View name** choice list on the list configuration page. Ensure the **List view** shows the list to be modified.

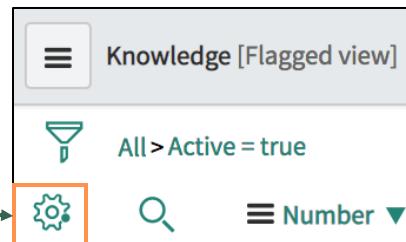
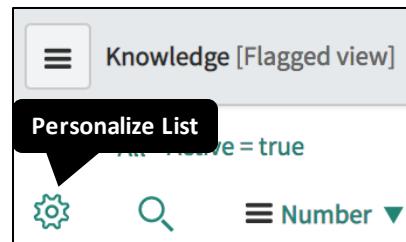
## Views: Personalization (logged-in user)

Personalize a list to show or hide fields on a view, as well as change the list column order for the current logged-in user.

Click the **Personalize List** (gear) icon in the list column header.

Use the **slushbucket** to add/remove/reorder columns for the personalized list and save.

The Personalize List (gear) icon changes to show the list has been personalized.



Where the List Layout configuration affects everyone\*, the **Personalize List** modifies the layout of a list for an *individual user*. It does **not** affect the platform default.

\*except individuals using personalized layouts for the view created using Personalize list

The following can be done through Personalize List Columns:

- **Add Columns:** In the available section, select the columns to add and select 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

# Finding Information

Find information quickly in ServiceNow with any of the available search functions:

- **Wildcards**
- Phrase Searches
- Searching Lists
- Boolean Operators
- Attachment Searches
- International Character Sets
- Punctuation

Wildcard Syntax	Search Type
<code>*searchTerm</code>	contains
<code>!*searchTerm</code>	does not contain
<code>searchTerm%</code>	starts with
<code>%searchTerm</code>	ends with
<code>=searchTerm</code>	equals
<code>!=searchTerm</code>	does not equal
<code>searchTerm</code>	greater than or equal to

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. Various wildcards can be used to refine the 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.



**Time**  
**15-20m**

- Create a new Infinity list view on the Incident table
- Apply and save a filter on a list of records
- Locate and update incident records using inline editing

## Lab 1.2: Lists and Filters

As the **Service Desk Manager**,

I need an **Infinity specific list view of incidents**

So senior resources can **limit time spent finding and updating** Infinity incident records.

# Lists and Filters

Lab

1.2

⌚15-20m

## Lab Objectives

You will achieve the following objectives:

- Create a new Infinity list view on the Incident table
- Apply and save a filter on a list of records
- Locate and update incident records using inline editing

## Scenario

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.

### A. Create the Infinity List View

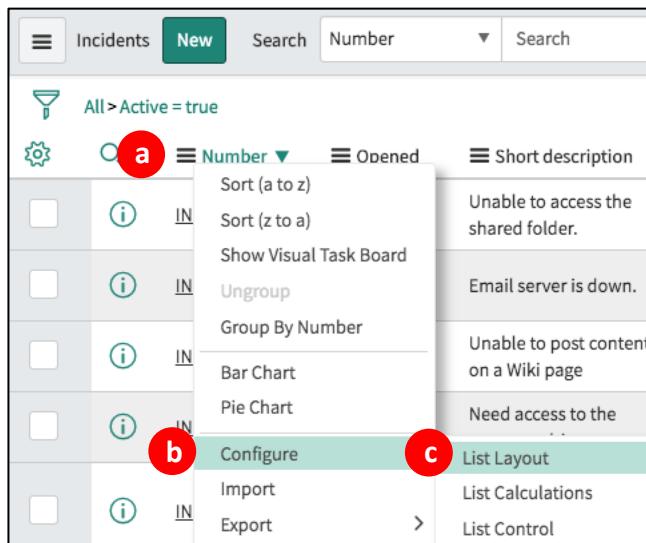
The system administrator user has the appropriate permissions for creating a new list view on incident – we will assume they have already received the requirements from Winnie Reich.

1. Impersonate **System Administrator** to complete the following steps.
2. **Incident > Open**

**Note:** You may want to add the **Incident > Open** module as a favorite!

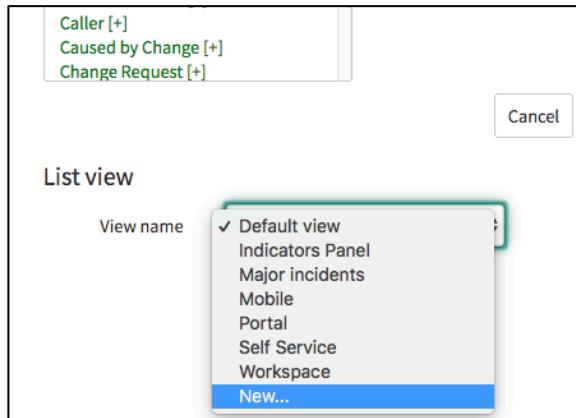
3. From the **List Layout**, the slushbucket will be used to create a new list view.  
Complete the following steps:

- Select a **Column Context Menu** (any column context menu will work)
- Select **Configure**
- Select **List Layout**



**Note:** Selecting any column context menu will work; Number was used in this example

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



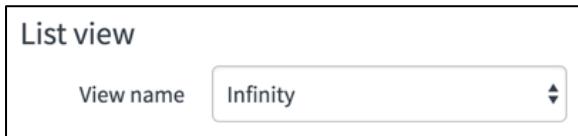
**Note:** Select **New...** to create a new list view. Choosing an existing list view will allow you to modify it.

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



7. Select **OK**.

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



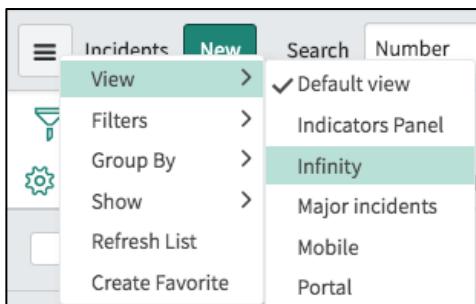
8. Working with the **Available** and **Selected** buckets, use the **Add** and **Remove** buttons (">" and "<" icons, respectively) to create the Infinity list view with the following fields:

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

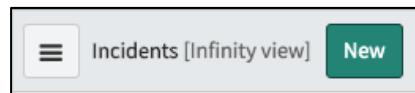
**Note:** Verify the fields are listed in the order shown at the left. Use the Move-up and Move-down arrows (at the right of the Selected box) to set the correct order.

**TIP:** Multiple fields may be selected using the Ctrl (or Command) key and moved or removed with one click.

9. Once the fields have been added to the Selected bucket, click the **Save** button.
10. Impersonate **Kevin Edd** to confirm the view is available for the **Service Desk** group.
11. As Kevin Edd, navigate to **Incident > Open**.
12. Open the **List Context Menu** and select **View**, then finally select **Infinity**:



**Note:** The list view name now appears at the top of the list in square brackets after the table name

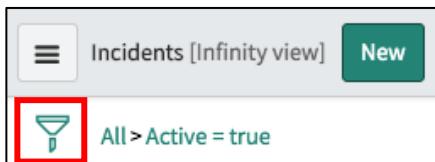


13. Confirm the fields appear in order, from left to right, as listed in step 8 above.

## B. 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):

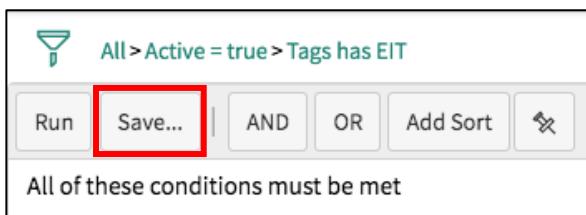


2. Add the following **AND** condition: **Tags | EIT**

A screenshot of the filter condition builder. The top bar shows a funnel icon and the condition "All &gt; Active = true". Below the bar are buttons for "Run", "Save...", "AND", "OR", "Add Sort", and a delete icon. A note says "All of these conditions must be met". Underneath, there's a row for "Active" with dropdowns for "is" and "true". Below it, a new condition "Tags | EIT" is being added, with its input field also redboxed. Buttons for "AND" and "OR" are visible to the right.

**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 internal testing incidents from customer incidents – it stands for **Employee Infinity Testing**.

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...**

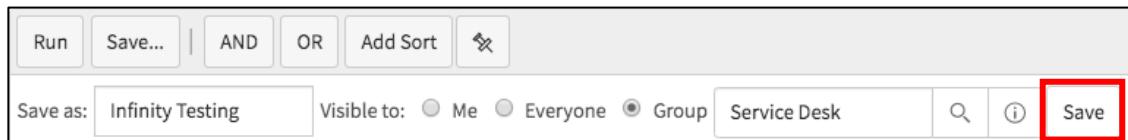


6. Enter **Infinity Testing** into the **Save as** field.
7. Next, select **Group** for **Visible to**.

**Note:** The ability to select a group to share with others is provided by additional user permissions. For this exercise, the Service Desk group was granted the **filter\_group** role.

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

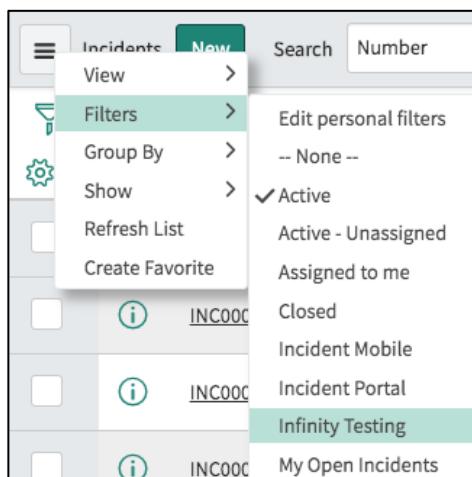
9. Select **Save**:



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 Context Menu** and select **Filters**, then finally select **Infinity Testing**:



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

Incidents <span>New</span> Search Number ▾					
<span>Filter</span> All > Active = true > Tags has EIT					
		Number ▾	Opened	Short description	Caller
	<span>i</span>	<a href="#">INC000042</a>	2017-09-15 02:31:21	password reset request	Megan Burke
	<span>i</span>	<a href="#">INC000023</a>	2017-09-15 02:33:36	Infinity showing an error - employee testing	Ted Keppel

## C. Locate a Missing Incident

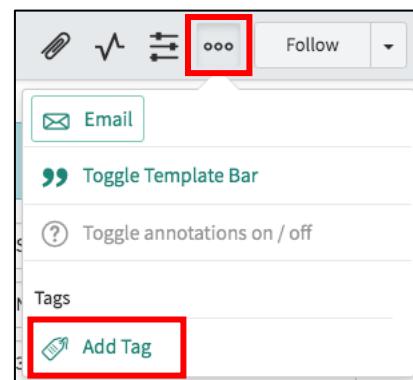
Winnie Reich has emailed Megan Burke to report an incident submitted by another employee that did not follow the current **EIT** tagging convention.

**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. As **Megan Burke**, using methods of your choice, locate and open Alissa's incident record: **INC0000061**.

**HINT:** Try performing a global search for **INC0000061**.

2. With the **INC0000061** record displayed, open **More options** from the form header:
3. Click **Add Tag**.
4. Type **EIT** into the **Add tag...** field.



5. Press **Enter** on your keyboard to add the tag to the incident record:
6. **Update** the record.
7. Apply the **Infinity Testing** list view.

**Note:** By updating the record, you should return to the filtered incident list. If not, navigate to Incident > Open and then apply the **Infinity Testing** filter.

8. Confirm all three Infinity testing records display:

		Number ▼	Opened	Short description	Caller
<input type="checkbox"/>		<a href="#">INC0000061</a>	2017-09-15 02:32:43	Infinity holographic settings page will not display	<a href="#">Alissa Mountjoy</a>
<input type="checkbox"/>		<a href="#">INC0000042</a>	2017-09-15 02:31:21	password reset request	<a href="#">Megan Burke</a>
<input type="checkbox"/>		<a href="#">INC0000023</a>	2017-09-15 02:33:36	Infinity showing an error - employee testing	<a href="#">Ted Keppel</a>

## D. Update Infinity Incident Records

Now that all Infinity incident records are accounted for, Winnie Reich has asked Megan Burke 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**:

<a href="#">INC0000042</a>	5 - Planning	New	<a href="#">Megan Burke</a>	<b>Database</b>	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 type="button" value="Inquiry / Help"/>	▼	
Subcategory:		
<input type="button" value="-- None --"/>	▼	

<a href="#">INC0000042</a>	5 - Planning	New	<a href="#">Megan Burke</a>	<b>Inquiry / Help</b>	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	5 - Planning	New	Alissa Mountjoy	a Network
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help
INC0000023	5 - Planning	New	Ted Keppel	b 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:

Subcategory:

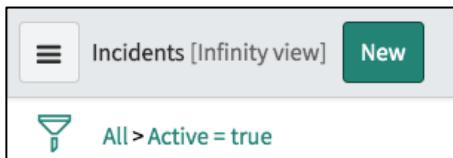
2 rows will be updated

- Use the **Category** drop-down to select **Software**.
- Save** (Enter) to update both records.
- Your Infinity incident list should look like the following:

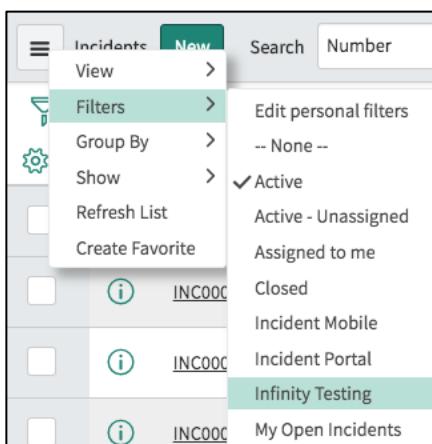
≡ Number ▼	≡ Priority	≡ State	≡ Caller	≡ Category
INC0000061	5 - Planning	New	Alissa Mountjoy	Software
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help
INC0000023	5 - Planning	New	Ted Keppel	Software

## Lab Verification

1. Impersonate Kevin Edd
2. Incidents > Open
3. Verify the *Infinity* list view is visible



4. Verify the *Infinity Testing* filter is available



5. Verify there are three Infinity records (Tag is EIT)

		Number	Opened	Short description	Caller
	(i)	INC0000061	2017-09-15 02:32:43	Infinity holographic settings page will not display	Alissa Mountjoy
	(i)	INC0000042	2017-09-15 02:31:21	password reset request	Megan Burke
	(i)	INC0000023	2017-09-15 02:33:36	Infinity showing an error - employee testing	Ted Keppel

**Congratulations on completing the Lists and Filters lab!**

**Now you understand how to create/edit list views and use filters to provide information in an order that makes it easier to use!**

## Knowledge Check

What is a collection of *permissions* used to grant access to different parts of the platform?

- **Role**

What is a common way of testing functionality for different roles?

- **Impersonate user**

A row represents what in the ServiceNow database?

- **Record**



What are the three different types of context menus in a list view?

- **List, Column, and Record**

What are the three components of a filter condition?

- **Field**
- **Operator**
- **Value**

What defines which fields appear in a list and the order of said fields?

- **View**

## **Section 1.3:** **Form Configuration**

### User Story

---

As the **Service Desk Manager**,

I need **testing-specific information on the incident form**

So we can **maintain the quality of data needed to accurately identify and resolve employee testing issues.**



## Viewing Single Records: Forms

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

The screenshot shows a ServiceNow user form for a record named 'User = Beth Anglin'. The form contains the following fields:

- First name: Beth
- Last name: Anglin
- Title:
- Department: Sales
- Location: 6304 Northwest Barry Road, Kansas City, MO
- Time zone: System (America/Los\_Angeles)

Below the form are two buttons: **Update** and **Delete**.

**Related Links**

- [View Subscriptions](#)
- [Reset a password](#)

At the bottom, there is a navigation bar with tabs: Entitled Custom Tables, Roles (28), Groups (6), Delegates, Subscriptions, and Manage Subscriptions. The 'Groups' tab is selected. Below the tabs is a search bar and a list of groups:

- User = Beth Anglin
- Group
- Network CAB Managers
- Hardware

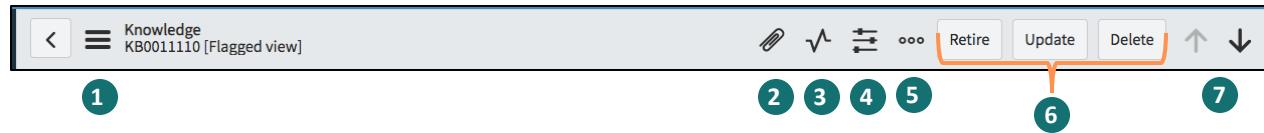
A form displays information from one record in a table. The specific information depends on the type of record displayed. Users can view and edit records in forms. Administrators can configure what appears on forms.

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.

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

## Header Icons

now.



1. **Form Context Menu:** Options related to viewing and filtering form data
2. Add or remove files with **Manage Attachments**
3. Show the **Activity Stream** for a real time, sequential display of record activities
4. **Personalize form:** show/hide fields on the form for the selected view
5. Open the **More Options** menu for additional form tools like templates and tags
6. Form UI actions (e.g. Retire, Update, Delete, Save, etc.)
7. Cycle through records by using **Next record** and **Previous record**

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

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

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

# Field Types

- 1. Reference** - Query that displays records from another table
- 2. Date/Time** - Populated with the day and time of day
- 3. String** - Freely populated using letters, numbers, and special characters
- 4. Choice** - Drop-down menu with multiple values
- 5. True/False** - Boolean field that appears as a checkbox

The screenshot shows two ServiceNow forms. The top form is a general article creation form with fields like Number, Knowledge base, Category, Published, Valid to, Short description, Article type, Workflow, Source Task, Attachment link, and Display attachments. The 'Category' field is highlighted with a red box and an arrow points down to its corresponding record in the bottom form. The bottom form is a 'Knowledge Category' record for 'Best Practices', showing fields for Label (Best Practices), Value (best-practices), Parent ID (Knowledge Base: Insta), and Active (checked). There is also an 'Update' button.

Forms include various field types, each with unique attributes.

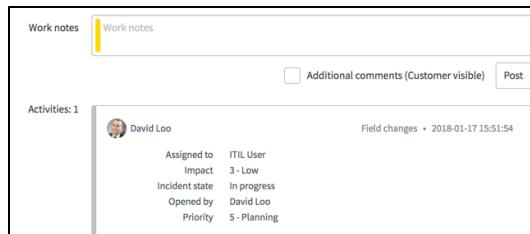
Some common field types include:

- 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.

# Formatters

## Formatter

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



## Formatter Examples

Some Formatters included in the base platform are:

- Activity
- Process
- Parent breadcrumbs
- Approval summarizer
- CI relations



The **Activity Stream** is an example of a formatter; it displays a list of activities

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

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

# Form Designer

Quickly create new or change existing form layouts

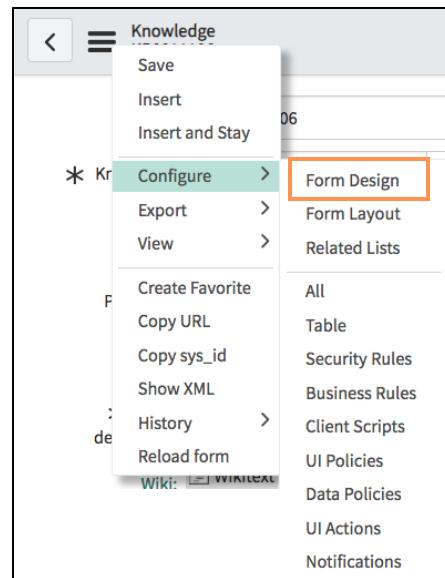
- Change the order/location of fields on a form for different form views
- Add fields to appear on the form
- Configure field properties

To access the Form Designer

- Open **Form Context menu**
- Select **Configure > Form Design**

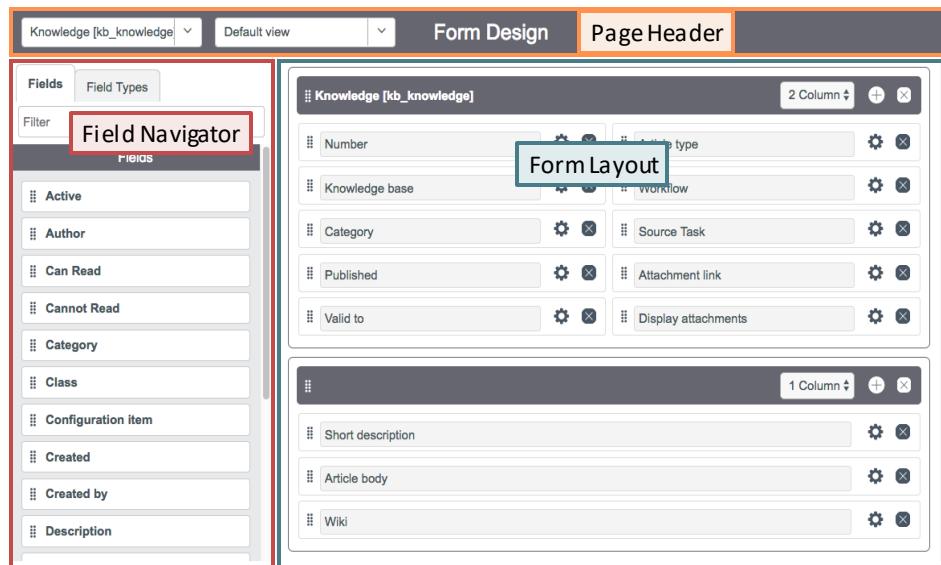


The Form Designer opens in a separate tab.



**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.

# Form Design Interface



There are 3 main components of the form designer:

- **Page header:** select the table and view from the page header. Other actions, Undo and Save, also appear on the page header after changes have been made to the form
- **Field navigator**
  - **Fields** tab - existing fields for the selected table which can be added to the form
  - **Field Types** tab - field types which can be added to the form. Adding a field to the form this way adds a field to the selected table
  - **Filter:** allows you to search fields or field types
- **Form layout:** graphic representation of the fields, sections, and other elements on the form

Fields can be dragged and dropped to different locations on the form and new fields can be added to the form by dragging and dropping from the Fields tab or the Field Types tab.

Documentation regarding the form designer can be found by searching ServiceNow's product documentation ([docs.servicenow.com](https://docs.servicenow.com)) for the article titled **Using the form designer**.

# Form Designer: Sections

now.

Sections are used to break up the form and group like information together.

Sections can be:

- One or two columns
- Added to a form (+)
- Removed from a form (X)\*

The screenshot shows the Form Designer interface for a table named 'Knowledge [kb\_knowledge]'. The form is set to '2 Column' layout. It contains ten fields arranged in two columns. The left column includes 'Number', 'Knowledge base', 'Category', 'Published', and 'Valid to'. The right column includes 'Article type', 'Workflow', 'Source Task', 'Attachment link', and 'Display attachments'. Each field has a gear icon for configuration and a delete icon (X).

\*By default, the first section on the form is read-only. That is, the label cannot be changed (contains the name of the table) and the section cannot be removed.

# Form Designer: Fields

now.

Each field is represented by a cell on the form which contains (left to right):

- Handle icon
- Field label
- Settings icon
- Remove icon

Knowledge [kb_knowledge]	
2 Column	
Number	Article type
Knowledge base	Workflow
Category	Source Task
Published	Attachment link
Valid to	Display attachments

Cell components:

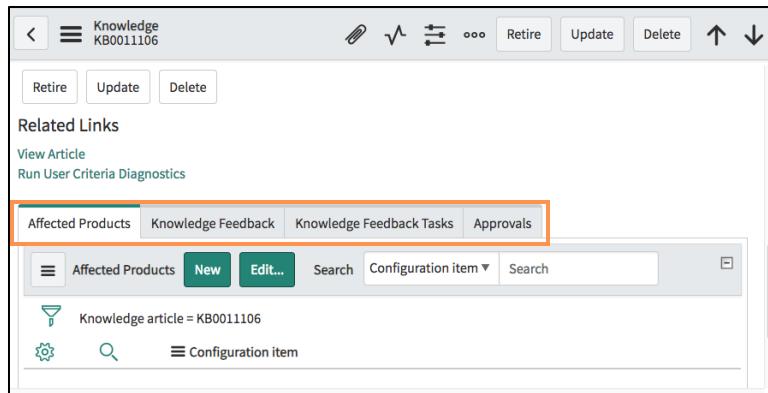
- **Handle icon:** use for moving the field to a different location on the form
- **Field label:** read-only to identify the field
- **Settings icon:** use to modify the field settings (label, mandatory, read only, reference, choices, etc.)
- **Remove icon:** use to remove the field from the form (not from the table)

## Related Lists

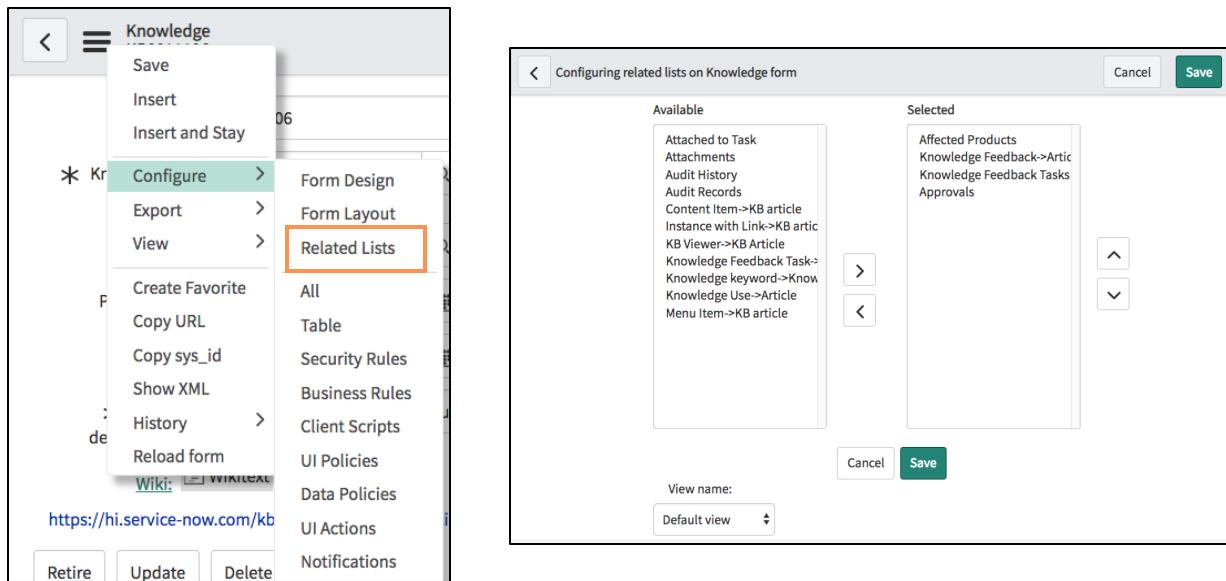
Related lists show records in tables that have a relationship to the current record. Related lists are presented as tabs at the bottom of the form view.

From a related list, a user can select records using the **Edit...** button or create new records using the **New** button.

As with other lists, a user can personalize the columns or filter the list using the **gear icon** or the **funnel icon**.



## Form Configuration: Related Lists



Using the slushbucket, related lists can be added, removed, or moved on the form view.

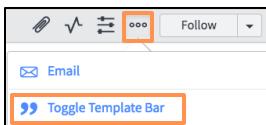
- To *add* a related list to the form, locate the list item in the Available column and double-click the list item to move it to the Selected column.
- To *remove* a related list from the form, locate the list item in the Selected column, then double-click the list item to move it to the Available column.
- To *change the display order* for the related lists, click on the list item in the Selected field, then use the arrows at the right to move the item up or down in the list. Top to bottom in the Selected column corresponds to tabs being shown left to right in the related lists section of the form.

**NOTE:** Ensure the **View name** is correct for the form you are modifying.

# Templates

**Templates** allow fields to be populated automatically, simplifying the process of submitting new records

Click the **More options** icon (  ) from the form header, then **Toggle Template Bar** to work with templates



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

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 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 are used frequently, such as incident, problem, and change. There is no limit to the number of templates that a user 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 by-pass process, like mandatory fields, UI policies, etc. This is especially important for any record using condition based workflows.

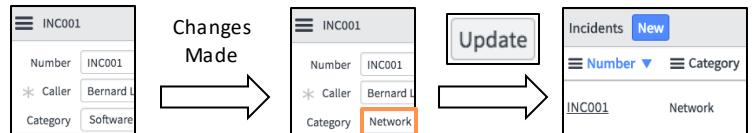
# Saving Forms

Save records by using one of the following methods:

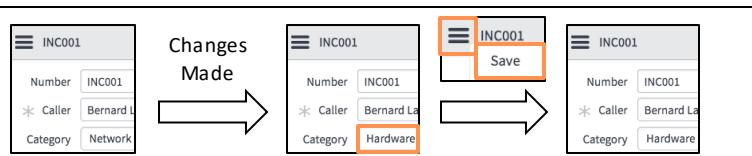
Click **Submit** to save changes on a new form and return to the previously viewed page



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



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



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 with certain record types.

**Additional Comments** are visible to all users accessing the record, whereas **Work Notes** are visible to only users with the **itil** role.

**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.

Once enabled, select the Form Context 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. Insert will exit the form returning to the previously viewed page, but Insert and Stay will remain on the form.

There is no "Save As" in ServiceNow but **Insert** closely emulates this functionality.



**Time**  
**15-20m**

- Create and configure a new form view using the Form Designer
- Create and update Infinity incident records

## Lab 1.3: Form Configuration

As the **Service Desk Manager**,

I need **testing-specific information on the incident form**

So we can **maintain the quality of data needed to accurately identify and resolve employee testing issues.**

# Form Configuration

Lab

1.3

⌚15-20m

## Lab Objectives

You will achieve the following objectives:

- Create and configure a new form view using the Form Designer
- Create and update Infinity incident records

## Scenario

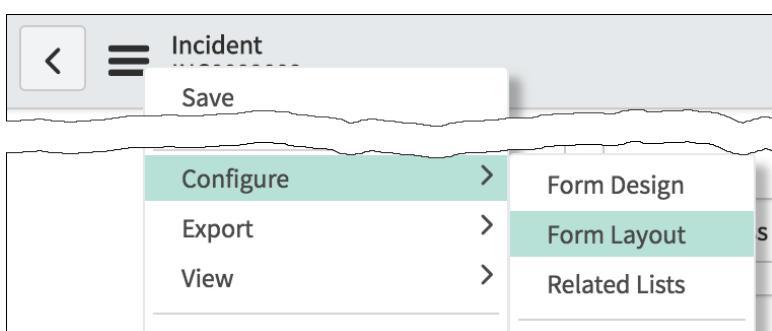
Internal employee testing of Infinity has proven worthwhile for a number of reasons.

Winnie Reich will lead an initiative to further improve and organize Infinity testing support by requesting a new form view on the incident table that contains appropriate fields for accurately identifying reported issues.

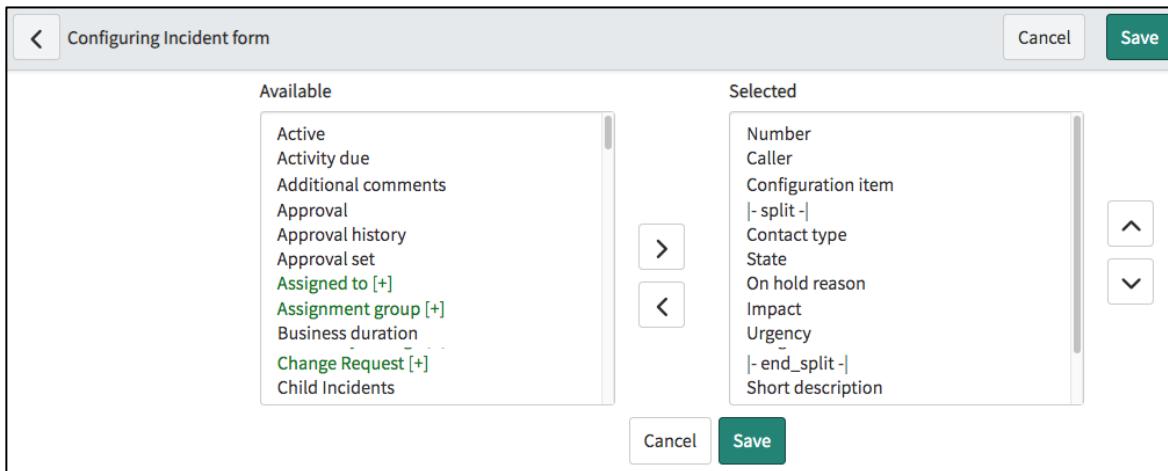
 **Lab Dependency:** Requires the completion of Lab 1.2

### A. Create the Infinity Form View

1. Verify you are logged in as the **System Administrator**.
2. **Incident > Open**
3. Open the record for **INC0000061**
4. Open the **Form Context Menu** and select **Configure**, then finally select **Form Layout**:



The Configuring Incident form page displays:



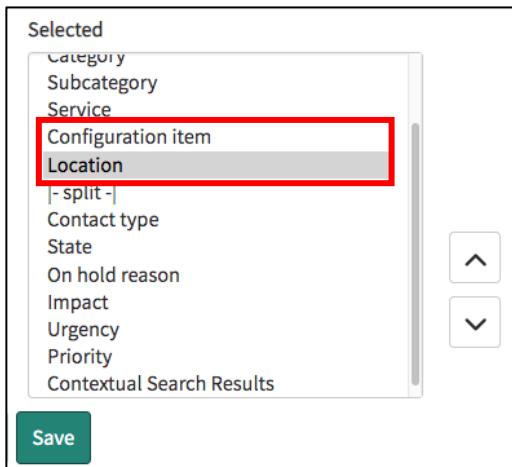
5. Find and highlight the **Location** field under the Available list:

The screenshot shows the 'Configuring Incident form' dialog. Under 'Available', the 'Location [+]' field is highlighted with a red rectangle. To the right, a note says: *Note: The Location field may not be where you expect it to be. Try highlighting the first field in the Available column and start typing Location to find it quickly*.

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

The screenshot shows the 'Configuring Incident form' dialog. Under 'Available', the 'Location [+]' field is selected and highlighted with a red rectangle. Between the 'Available' and 'Selected' lists is an 'Add' button (a right-pointing arrow) which is also highlighted with a red rectangle. To the right, a note says: *Note: Adding a field to the form this way adds it to the bottom of the Selected list. To move the field, the Move up and Move down buttons must be used.*

7. Use the **Move up** button (^) to move the **Location** field under **Configuration item**.



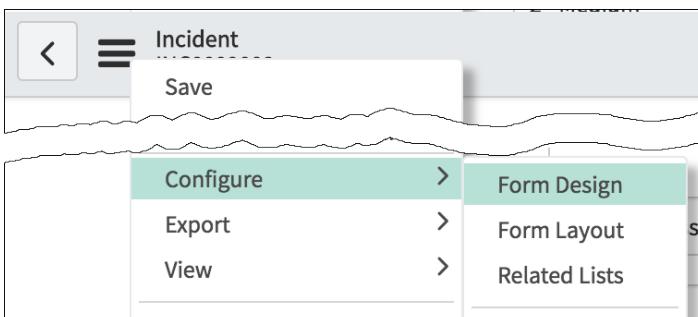
8. Select **Save** to return to the incident form.
9. Verify **Location** appears under the **Configuration Item** field:

A screenshot of an incident form. It includes fields for Category (Software), Subcategory (-- None --), Service (search bar), Configuration item (search bar), and Location (search bar). The "Configuration item" and "Location" fields are highlighted with a red box.

## B. Modify Form View Using Form Designer

While **Form Layout** can be used to modify the form view, the **Form Designer** offers an improved experience. Because of its graphical user interface, it makes it easier to visualize the form view's end result. Several additional configuration options are available in **Form Design**.

1. Open the **Form Context Menu** and select **Configure**, and then **Form Design**.



2. The **Form Designer** will open in a new tab or window. The page should look like this:

**Note:** At the top left of the page are two drop-down menus in the header; the menu on the left indicates the table the form view is associated with, and the menu on the right includes the various views defined for the selected table.

3. Open the **view** (right) menu and select **New...** at the bottom of the list.

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

Select **OK**.

**Note:** The new **Infinity** view is automatically selected in the view menu on the page's header:

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



- 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**.
- Drop the field in the desired location on the form.



- 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.

- Find the **Service** field on the form layout, then click the **Remove field Service** (X icon) to remove it from the view:



**Note:** The X icon name changes based on the field. In this example, the tooltip text shows **Remove field Service** when hovering over the X icon. For the caller field, the tooltip text shows **Remove field Caller**.

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

9. Click and drag the **Contact type** field to be displayed beneath the **Number** field:

The screenshot shows the ServiceNow Field Navigator interface for the 'Incident [incident]' form. The 'Fields' tab is selected. A red border highlights the 'Number' field, and a red arrow points from it to its current position above the 'Contact type' field. The 'Contact type' field is also highlighted with a red border. Other fields visible include 'State', 'Caller', 'Created by', and 'On hold reason'. Each field has a gear icon for settings and a close button.

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

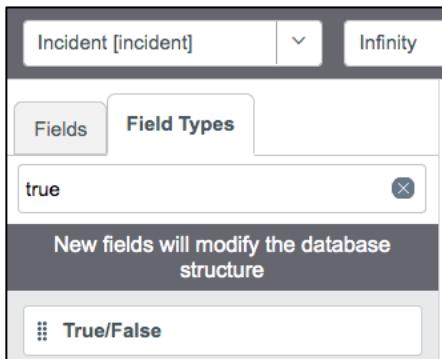
The screenshot shows the ServiceNow Field Navigator interface for the 'Incident [incident]' form. The 'Fields' tab is selected. The fields are now arranged in a grid layout. The first column contains 'Number', 'Contact type', 'Caller', 'Location', 'Category', 'Subcategory', and 'Configuration item'. The second column contains 'State', 'On hold reason', 'Impact', 'Urgency', 'Priority', 'Assignment group', 'Assigned to', 'Created by', 'Updated', and 'Updated by'. Each field has a gear icon for settings and a close button.

## C. Define a New Field

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

The screenshot shows the ServiceNow Field Navigator interface. The 'Field Types' tab is selected. A search bar labeled 'Filter' is present. Below it, under the 'Journal' category, there is a list of field types: 'List'. Each item has a gear icon for settings and a close button.

2. Scroll down to find the **True/False** field type or search for it using the Filter.



3. Add the field to the form layout under the **Caller** field:



4. Click the **Edit field New True/False** (gear icon) to configure the field's properties.



5. 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.

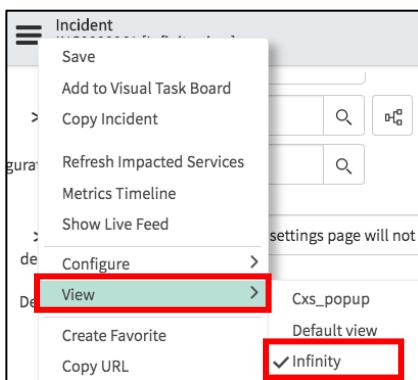
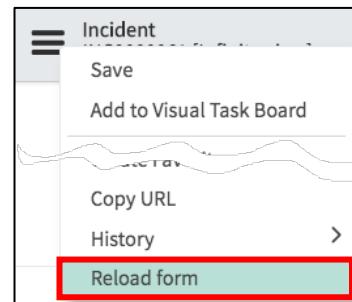
6. Close the **Properties** window by clicking the close icon (x).
7. Select the **Save** action from the page header to save the form view:



- Close the **Form Design** tab/window and return to the ServiceNow lab instance.

The screenshot shows the ServiceNow Incident form interface. At the top, it displays the record number 'INC0000061'. Below the header are several input fields: 'Number' (INC0000061), 'Contact type' (Self-service), 'State' (New), 'Category' (Software), 'Impact' (3 - Low), 'Subcategory' (None), 'Urgency' (3 - Low), 'Service' (empty), and 'Priority' (5 - Planning). There are also search and filter icons next to some of the input fields.

- Select **Reload form** from the **Form Context Menu** to reload the form and update the View options
- Open the **Form Context Menu**, select **View**, and select **Infinity** to display the form as designed!



## D. Create and Update Incident Records

With the new form view defined, the next steps are to update Infinity testing records by inputting the information into the correct fields. These steps will be completed by Kevin Edd.

Kevin Edd also received a new incident from Buster Wubbel in person. Submit a new incident to capture the details provided by Buster Wubbel.

- Impersonate **Kevin Edd**.
- Incident > Create New**
- Verify the *Infinity* form view is displayed and complete the form using the following:

Contact type: **Walk-in**

Caller: **Buster Wubbel**

Employee: [checked]

Category: Software

State: In Progress

Assignment group: Service Desk

Short description: Issue discovered with two step authentication

Description: Authentication requirements for logging in to Infinity are not working as expected.

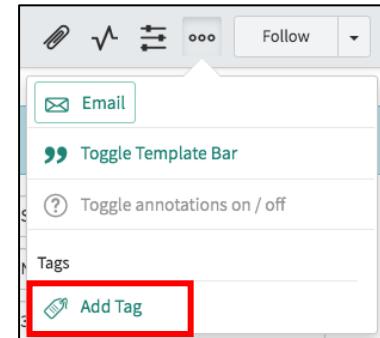
**Note:** The **Location** field should have populated automatically because of the **Caller** value input.

- Choose **Save** by opening the form context menu or by right-clicking the form header.



- Open **More options** from the form header, select **Add Tag**.

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



- Update the record

- 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 testing incident records. (The incident number may be different than shown):

Incidents [Infinity view]										
All > Active = true > Tags has EIT										
	Number	Priority	State	Caller	Category	Subcategory	Short description	Assignment group	Assigned to	Tags
<input type="checkbox"/>	INC0010001	5 - Planning	In Progress	Buster Wubbel	Software		Issue discovered with two-step authentication	Service Desk	(empty)	 EIT 
<input type="checkbox"/>	INC0000061	5 - Planning	New	Alissa Mountjoy	Software		Infinity holographic settings page will not display	Infinity Customer Support	(empty)	 EIT 
<input type="checkbox"/>	INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help		password reset request	(empty)	(empty)	 EIT 
<input type="checkbox"/>	INC0000023	5 - Planning	New	Ted Keppel	Software		Infinity showing an error - employee testing	Infinity Customer Support	(empty)	 EIT 

**Challenge: (REQUIRED)** Update records to change the state, assignment group, and “Employee” checkbox values as shown in the table, below.

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

Number	Employee	State	Assignment group
INC0000023	True	In Progress	Service Desk
INC0000042	True	In progress	Service Desk
INC0000061	True	On Hold	Service Desk

**NOTE:** If required, set the value for the **On hold reason** and **Additional Comments (Customer visible)** fields to **Awaiting Caller**.

## Lab Verification

1. Impersonate **Megan Burke** (a member of the Service Desk group)
2. Navigate to **Incident > Open** and apply the *Infinity Testing* list filter.
3. Verify there are four (4) open Infinity Incidents:

	Number	Priority	State	Caller	Category	Subcategory	Short description	Assignment group	Assigned to	Tags	Updated
<input type="checkbox"/>	INC0010001	5 - Planning	In Progress	Buster Wubbel	Software		Issue discovered with two step authentication	Service Desk	 EIT		2018-02-26 09:09:43
<input type="checkbox"/>	INC0000061	5 - Planning	On Hold	Alissa Mountjoy	Software		Infinity holographic settings page will not display	Service Desk	 EIT		2018-02-26 09:15:49
<input type="checkbox"/>	INC0000042	5 - Planning	In Progress	Megan Burke	Inquiry / Help		password reset request	Service Desk	 EIT		2018-02-26 09:16:14
<input type="checkbox"/>	INC0000023	5 - Planning	In Progress	Ted Keppel	Software		Infinity showing an error - employee testing	Service Desk	 EIT		2018-02-26 09:13:53

4. Select **New** to create a new incident
5. Verify the Infinity view is available, and the format matches the screenshot, below:

Incident  
New record [Infinity view]

Number	INC0010002	State	New
Contact type	-- None --	Impact	3 - Low
* Caller	<input type="text"/>	Urgency	3 - Low
Employee	<input type="checkbox"/>	Priority	5 - Planning
Location	<input type="text"/>	Assignment group	<input type="text"/>
Category	Inquiry / Help	Assigned to	<input type="text"/>
Subcategory	-- None --	Created by	<input type="text"/>
Configuration item	<input type="text"/>	Updated	<input type="text"/>
* Short description	<input type="text"/>		
Description	<input type="text"/>		

**Challenge:** End the impersonation of Megan Burke using the User Menu

**Well done! You have created a new form view which can be used by the members of the Service Desk group!**

## Knowledge Check

Which field type displays as a check box on the form?

- **True/False**

Which field type displays information from another table?

- **Reference**

Which tool can be used to change the layout of the form view for all users?

- **Form Designer**



What are the three main components of the Form Designer?

- **Page Header**
- **Field Navigator**
- **Form Layout**

What allows fields to be populated automatically when applied?

- **Template**

Which UI action saves information for a new record and leaves the form?

- **Submit**

## Section 1.4: Branding

### User Story

---

As a **change enablement specialist**,

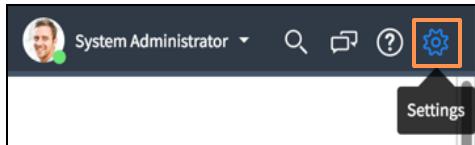
I want the **branding of our ServiceNow instance to align** with the rest of the organization

So **users have a seamless transition** to the ServiceNow platform and will more readily adopt the solution.



# Instance Personalization

Click the **Settings** icon from the banner frame to personalize your instance.



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

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

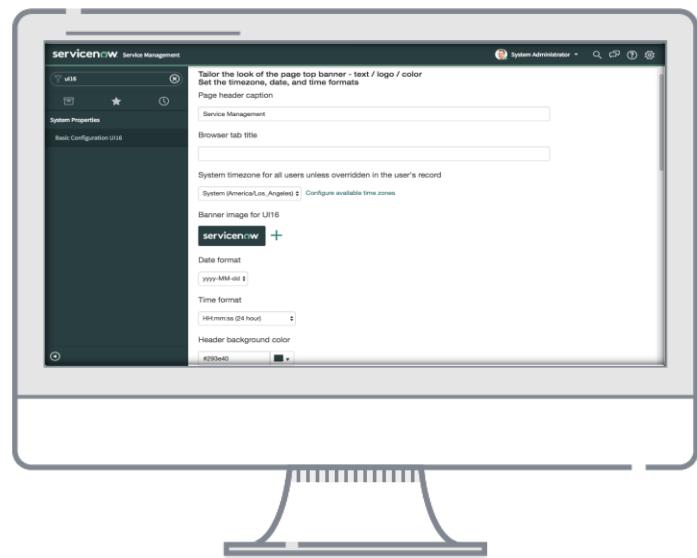
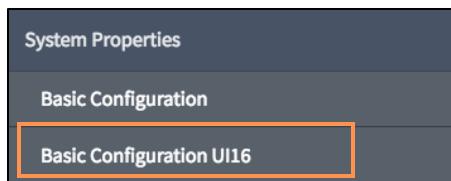
**NOTE:** Users may be limited to what settings they have access to based on their role.

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.

# Instance Branding

Many branding options are accessible from **System Properties > Basic Configuration UI16**



Different than personalizing an instance, configuring the system branding is something an administrator can do to change the look and feel of the instance for any user who logs in.

Customization and branding options include:

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

Features which can also be customized:

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



**Time**  
**10-15m**

- Use the UI16 module to brand the ServiceNow instance for all users

## Lab 1.4: Branding

As a **change enablement specialist**,

I want the **branding of our ServiceNow instance to align** with the rest of the organization

So **users have a seamless transition** to the ServiceNow platform and will more readily adopt the solution.

# Branding the Instance

Lab  
1.4

5-10m

## Lab Objectives

You will achieve the following objectives:

- Use the My Company module to update the logo for Cloud Dimensions
- Use UI16 module to update the colors to match the Cloud Dimensions brand for all users

## User Story

As a **change enablement specialist**, I want the **branding of our ServiceNow instance to align** with the rest of the organization so users have a seamless transition to the ServiceNow platform and will more readily adopt the solution.

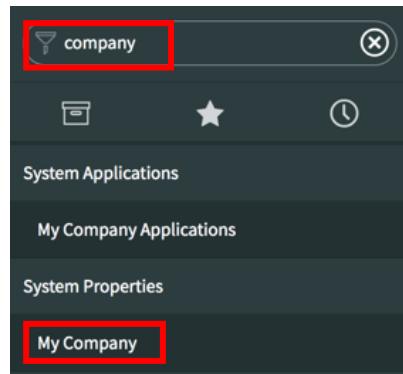
**Required Resource(s):** *CloudDimensions-CD-Logo-2DE8E4.png – available to download from Class Lab Files KB article*

### A. Locate the My Company module

*As the system administrator, you will modify the logo, text, and colors to match Cloud Dimensions branding. The image has been provided by the marketing team and was included in the class lab files downloaded previously.*

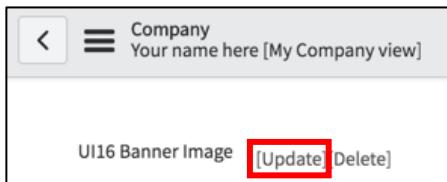
*The first step will be to modify the logo and banner text for the instance.*

1. As the System Administrator, enter the keyword **company** in the *Filter Navigator*

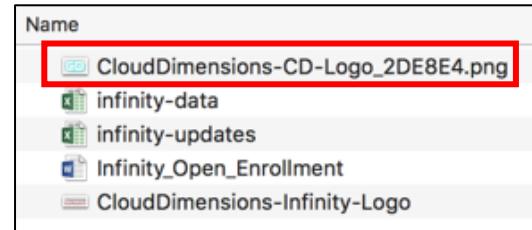
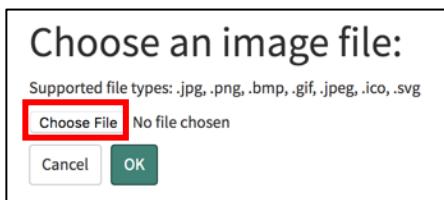


2. Select the **My Company** module within the *System Properties* application

3. Select **Update** for the *UI Banner Image*



4. Select **Choose File** to browse your computer and select the *CloudDimensions-CD-Logo\_2DE8E4.png* file



**Note:** To select the file, double-click the file name or click on the file name and select Open

5. Select **OK**
6. Enter **Cloud Dimensions <YourFirstName YourLastName>** in the **Banner Text** field

**Note:** Replace <YourFirstName YourLastName> with your own first and last name
7. **Update** the record and **refresh** the browser page
8. **Verify** the *UI banner image* and *banner text* have been updated



---

Well done! Now that the logo and instance name (banner text) have been updated, it is time to update the color scheme to match the Cloud Dimensions brand.

---

## B. Modify Instance using the UI16 Module

- As the System Administrator, use the Filter Navigator to locate the **Basic Configuration UI16** module. (**System Properties > Basic Configuration UI16**)



- Verify the fields on the form match the following (modify if necessary):

Page header caption: **Cloud Dimensions <YourFirstName YourLastName>**

Browser tab title: **Cloud Dimensions ServiceNow**

Banner image for UI16: **[CloudDimensions-CD-Logo\_2DE8E4.png]**

- Modify the fields according to the following color scheme:

**Note:** It may be easier to enter the color (e.g. #1d3341) in the first field in the list, then copy the number and paste it into the other fields with that color assignment.

As you enter values, especially for color fields, changes may or may not display in real-time. Some changes may not be visible until the form is saved, and the page refreshed.

Color	Applies to
#1d3341	<ul style="list-style-type: none"><li>- Header background color</li><li>- Navigation header/footer</li><li>- Navigation background expanded items</li><li>- Navigation selected tab background color</li><li>- Navigation unselected tab divider bar color</li><li>- Navigation separator color</li></ul>
#f8f6ef	<ul style="list-style-type: none"><li>- Banner text color</li><li>- Module text color for UI16</li><li>- Border color for UI16</li></ul>
#388ac7	<ul style="list-style-type: none"><li>- Header divider stripe color</li><li>- Unselected navigation tab icon and favorite icons color</li></ul>
#2de8e4	<ul style="list-style-type: none"><li>- Navigation selected tab divider bar color</li><li>- Currently selected Navigation tab icon color for UI16</li></ul>
#293e40	<ul style="list-style-type: none"><li>- Background for navigator and sidebars</li></ul>

- When finished, click the **Save** button.

System Configuration

Tailor the look of the page top banner - text / logo / color  
Set the timezone, date, and time formats

Page header caption

Cloud Dimensions

Browser tab title

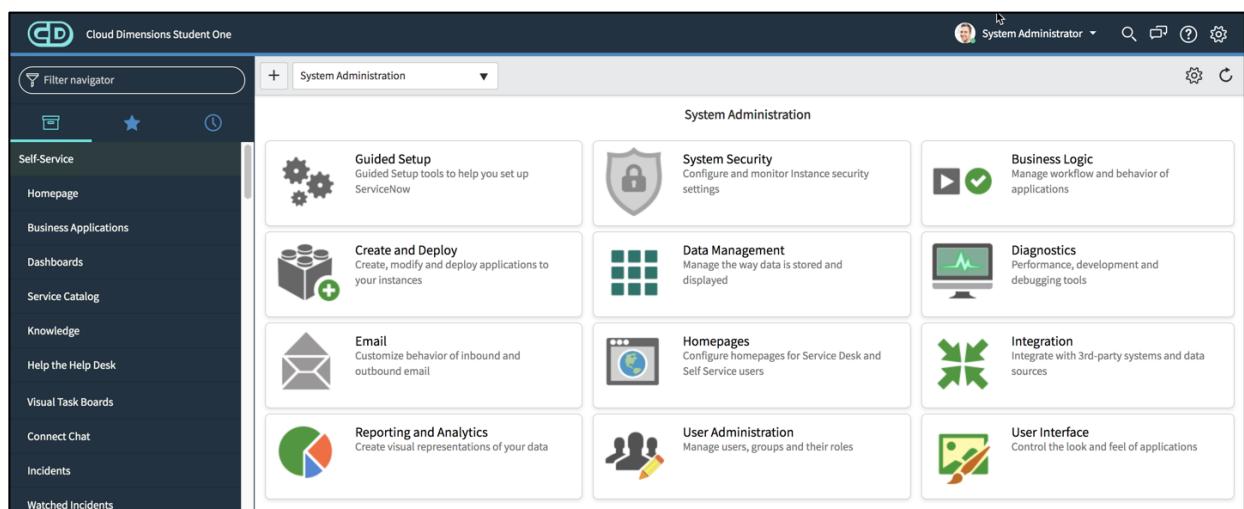
Cloud Dimensions ServiceNow

**Save**



## Lab Verification

- Verify the color scheme, banner, and text matches the *Cloud Dimensions System Theme*



Cloud Dimensions Student One

System Administration

Guided Setup	System Security	Business Logic
Guided Setup tools to help you set up ServiceNow	Configure and monitor instance security settings	Manage workflow and behavior of applications
Create and Deploy	Data Management	Diagnostics
Create, modify and deploy applications to your instances	Manage the way data is stored and displayed	Performance, development and debugging tools
Email	Homepages	Integration
Customize behavior of inbound and outbound email	Configure homepages for Service Desk and Self Service users	Integrate with 3rd-party systems and data sources
Reporting and Analytics	User Administration	User Interface
Create visual representations of your data	Manage users, groups and their roles	Control the look and feel of applications

**Congratulations!**  
**You have branded the instance to align with the rest of Cloud Dimensions!**

## Knowledge Check

Which icon on the banner frame is selected to customize an individual's instance?

- System settings (gear icon)

Which module allows an admin to change the look and feel for an instance?

- System Properties > Basic Configuration UI16



What are some reasons for changing the look and feel of a ServiceNow instance?

- To distinguish between production and non-production instances
- To comply with corporate branding and themes
- To increase user acceptance
- Any others?



## Module 2: Collaboration

User Interface and Navigation

### Collaboration

Database Administration

Self-Service and Automation

Introduction to Development

Capstone Project

### Module Objectives

- Explain how to access the service portal for the instance
- Understand how to submit an incident through the service portal
- Discuss when the My Work and My Groups Work modules should be used
- Demonstrate how to use work notes, additional comments, and connect chat to collaborate on a task
- Describe the differences between events and notifications
- Create a report by modifying an existing report using the Report Designer
- Share a report with a group

### Labs and Activities

- Lab 2.1: Task Management
- Lab 2.2: Notifications
- Lab 2.3: Reporting

## Section 2.1: Task Management

### User Story

---

As the **Service Desk Manager**,

I need to **support partner testers** with finding knowledge and the ability to submit incidents

So my team can more **quickly resolve unknown issues**.



# What is a ServiceNow Task?

## Tasks

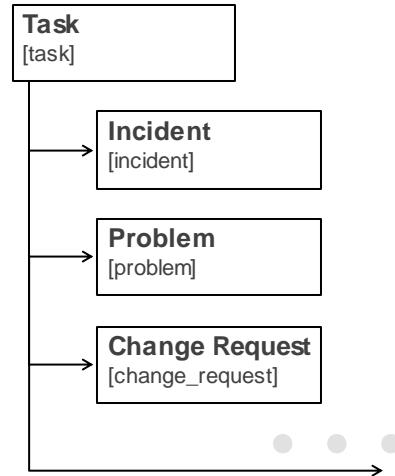


A **task** is any record that can be assigned or completed by a user in ServiceNow

A task record is created from a table that extends the **Task [task]** table

Work is performed upon a task and it is eventually moved to a closed state

Within ServiceNow, all work to be done is handled through tasks

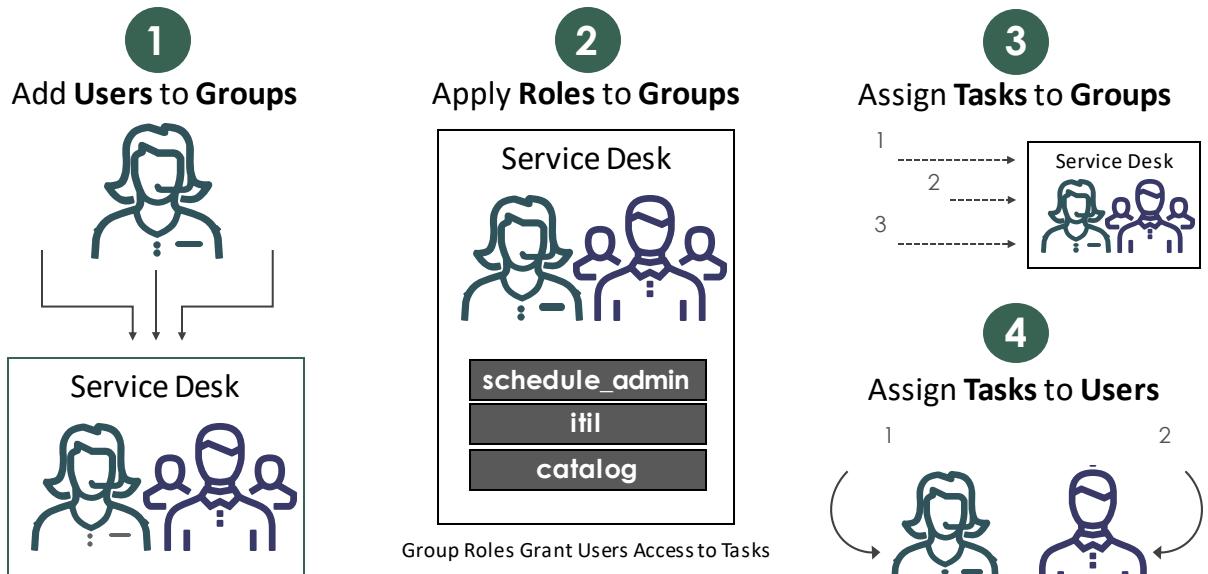


Tasks are created by users who are requesting the task to be performed, and are then updated as the task moves along a workflow. Tasks can be assigned to specific users or user groups.

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.

# Task Assignment



Users can belong to more than one group, and groups identify a subset of users based on roles.

Users in groups can be assigned permissions to:

- approve, change, or resolve incidents and requests
- provide a reference for alerts and notifications
- receive email notifications

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

- add 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.

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

With groups defined, tasks can be assigned to them and then to single users belonging to that group. In other words, a task record can be assigned to an assignment group and an assigned user.

# Task Assignment: Assignment Rules

Assignment rules can automatically set a value in the **assigned\_to** and **assignment\_group** fields of a task record

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

The screenshot displays the ServiceNow Assignment Rule configuration interface. It consists of two main sections: 'Applies To' and 'Assign To'.  
The 'Applies To' section contains:

- A dropdown menu for 'Table' set to 'Incident [incident]'.- 'Conditions' section with buttons for 'Add Filter Condition' and 'Add "OR" Clause'.- Operator and value selection fields: '-- choose field --', '-- oper --', and '-- value --'.

The 'Assign To' section contains:

- 'User' and 'Group' fields, each with a search icon.

Create an assignment rule 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. That is, 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 lookup rules is another type of assignment rule that can be created by navigating to the **System Policy > Rules > Assignment Lookup Rules** module. These rules only apply to incident records and have fewer options, compared to the other assignment rules.

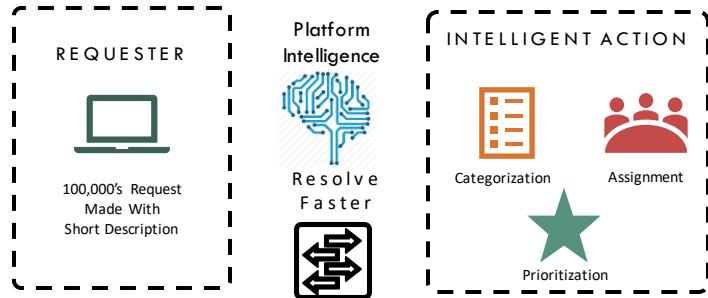
# Task Assignment: Platform Intelligence

now.

**Platform Intelligence** uses machine-learning algorithms to set field values during record creation

Improve efficiency and quality, and reduce cost by:

- Reducing task resolution times
- Reducing the number of interactions required to resolve tasks
- Reduce the error rates of categorizing and assigning work



*Learn more about Platform Intelligence in Now Learning*

Users with the **admin** or **ml\_admin** role will be able to initiate the setup of Platform Intelligence.

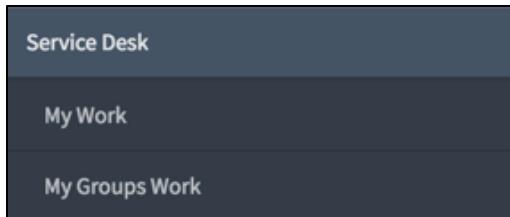
Once activated and defined, Platform Intelligence is truly personalized machine learning tailored to your data.

For example, when an incident is created, ServiceNow will automatically assign the correct category, priority and assignment group based on the record's short description. A requester no longer has to scroll down multiple lists to choose the most appropriate category. Additionally, the incident created gets the right attention and SLA it deserves, thus increasing the overall service level satisfaction.

This functionality does require multiple plugins – more information about setup and initial configuration can be found by searching [docs.servicenow.com](https://docs.servicenow.com).

# Accessing Tasks: My Work / My Groups Work

The Service Desk application menu allows you to locate all work assigned to your group(s) or to you



The **itil** role is required to access the **My Work** and **My Groups Work** modules in the Service Desk application

## My Work

List of all active tasks assigned to you, including:

- Change Request
- Group Approval
- Incident
- Knowledge Base Submission
- Request
- Security Case
- Visual Task Board Task

## My Groups Work

List of all active tasks assigned to your group(s) but not yet to an individual

There are various features for managing and closing task records.

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

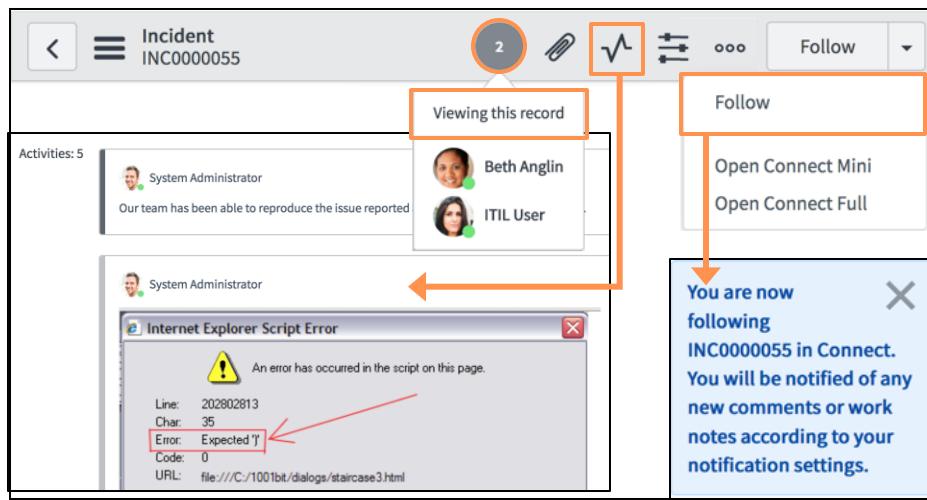
When an active task (which might include work such as incidents, problems, changes, and more) is routed to your group, it can be located under the **Service Desk > My Groups Work** module. From there, a group member or manager may assign the task to an individual within the group.

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

# User Presence

The **User Presence** feature facilitates synchronous collaboration within one record

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 **User 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.

## User Presence: Real-Time Editing

Edit records in real-time and see edits saved by other users, improving collaborative efforts

The screenshot shows a ServiceNow incident record for 'INC0000047'. The record includes fields such as Number, Caller, Category, Business service, Configuration item, Contact type, State, Impact, Urgency, Priority, Assignment group, and Assigned to. Two specific annotations are visible: a blue box around the 'Category' field containing the text 'ITIL User has modified this field value', and a red box around the 'Assigned to' field containing the text 'ITIL User has set this field to Bow Ruggeri'.

Real-time editing is an extension of User 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).

User Presence enables you to work on task records seamlessly with others using entirely different interfaces or devices to perform tasks. This is achieved when using Visual Task Boards (VTBs), Connect Chat, Apple Watch, and more.

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

## Notes Tab: Work Notes and Additional Comments

Use the **Notes** tab in a record to communicate to stakeholders and document task activities throughout the lifecycle

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: Activity

The **Activity** section located under the Notes tab provides a complete history of a record

It details:

- **Who** made an update?
- **What** was the update?
- **When** was the update made?

Selecting the filter (funnel) icon allows activity information to be filtered

**Who?**

Activities: 5

System Administrator  
Our team has been able to reproduce the issue reported and we are confirming a solution.

Additional comments • 2018-07-23 08:55:15

**What?**

Internet Explorer Script Error

An error has occurred in the script on this page.

Line: 202802813  
Char: 35  
Error: Expected ')' (at character 35)  
Code: 0  
URL: file:///C:/1001bit/dialogs/staircase3.html

Do you want to continue running scripts on this page?  
Yes No

**When?**

System Administrator  
Additional comments 2018-01-17 16:43:54

Filter Options (Sidebar):

- All
- Additional comments
- Assigned to
- Attachments
- Configuration item
- Impact
- Incident state
- Opened by
- Priority
- Relationship Changes
- Resolution code
- Resolution notes

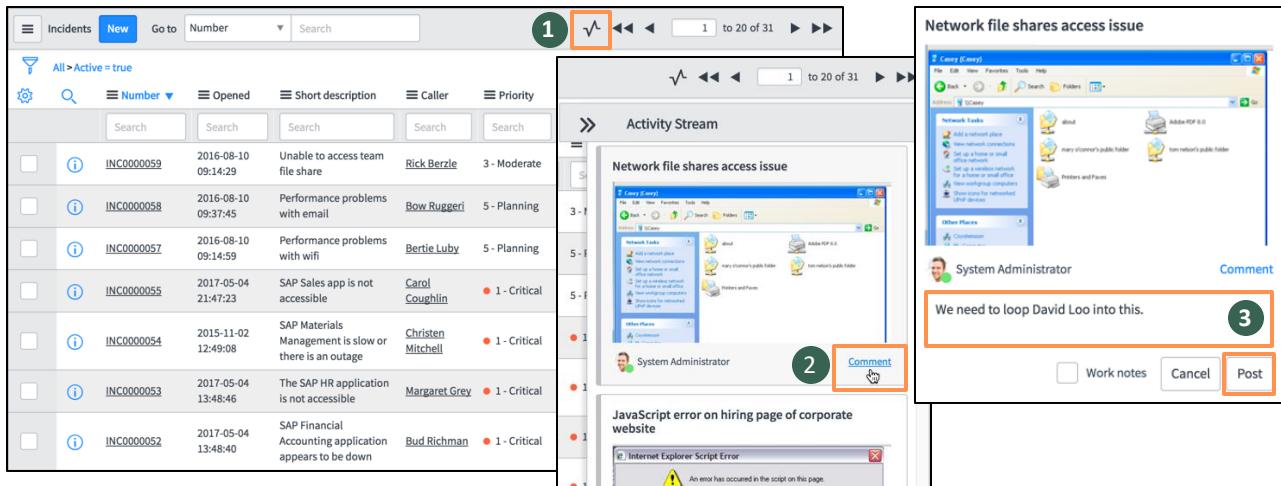
From creation through to closure, the entire history of an incident record is automatically tracked and recorded in the incident **Activity** section, located within the **Notes** tab.

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.

# Activity Streamline Editing

The **activity stream inline editor** enables users to contribute to actual work within a record without opening a form interface



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.

# Visual Task Boards

now.

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

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

The screenshot shows a 'Guided Board' titled 'Incidents by State'. It features four main lanes: 'New', 'In Progress', 'On Hold', and 'Completed'. Each lane contains multiple cards representing different incidents. The 'New' lane has four cards: 'Trouble getting to Oregon mail server', 'Can't access SFA software', 'Performance problems with email', and 'Performance problems with wifi'. The 'In Progress' lane has two cards: 'Network storage unavailable' and 'Please remove the broken hoffix from my PC'. The 'On Hold' lane has two cards: 'My desk phone does not work' and 'Can't log into SAP from my laptop today'. The 'Completed' lane has two cards: 'JavaScript error on hiring page of corporate website' and 'Internal Explorer Script Error'. Each card includes a small profile picture, the incident title, a brief description, and the creation date (e.g., '2mo ago'). At the top of the board, there's a 'Quick Panel' and a 'Taskboard tools' menu.

Use **Visual Task Boards (VTBs)** to create a 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 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 a Visual Task Board, navigate to **Self-Service > Visual Task Boards** and follow the displayed instructions for creating a board.



**Time**  
**10-15m**

- Browse the Service Portal
- Submit an incident from the Service Portal
- Use work notes, comments, and chat to cooperatively manage an incident

## Lab 2.1: Task Management

As the **Service Desk Manager**,

I need to **support partner testers** with finding knowledge and the ability to submit incidents

So my team can more **quickly resolve unknown issues**.

# Task Management

**Lab  
2.1**

⌚10-15m

## Lab Objectives

*You will achieve the following objectives:*

- Browse the Service Portal
- Submit an incident from the Service Portal
- Use work notes, comments, and chat to cooperatively manage an incident

## User Story

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

Before submitting an incident to report a problem, these partner users have been instructed to first browse the Knowledge Base in the Cloud Dimensions Service Portal.

If the user is unable to find a solution to their problem, they are instructed to submit an incident so they can receive support by an agent in the Infinity Customer Support group.

The Infinity Customer Support group has defined processes for incident assignment and resolution, which uses assignment rules and connect chat to ensure the best help is offered to an end-user.

## A. Browse the Service Portal

Start this lab by logging into the Cloud Dimensions Service Portal as a partner user experiencing problems during Infinity testing.

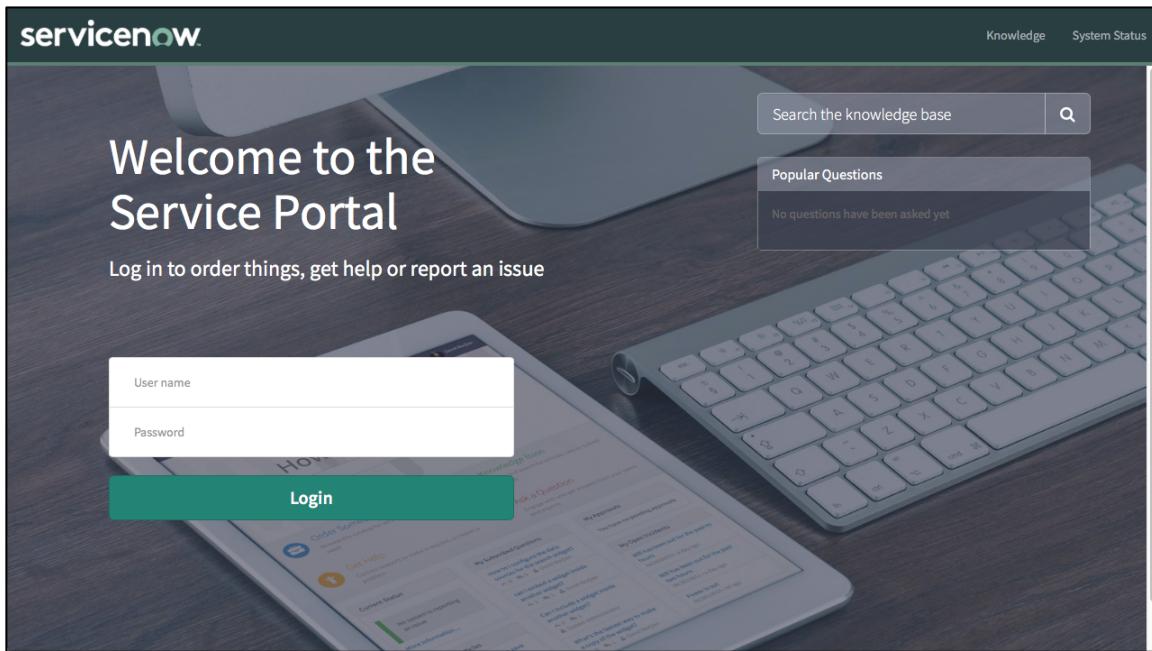
**TIP:** Open up another browser to complete these steps.

1. If you are 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 login screen:



4. Log in using the following user credentials:

**User name:** *jon.floyd*  
**Password:** *floydpass*

5. Type **infinity holograph broken** into the **How can we help?** search bar on the home page; then press enter on your keyboard.

**No Results** will display:

A screenshot of the ServiceNow Knowledge Base search results. The top navigation bar shows 'Home &gt; Search'. On the left, a sidebar has 'Sources' with options 'All', 'Knowledge Base', 'Questions and Answers', and 'Service Catalog'. Under 'Knowledge Base', the 'Knowledge Base' option is selected. The main content area shows a 'No Results' message: 'Your search - infinity holograph broken - did not match any documents'. Below this, there are two bullet points: 'Make sure all words are spelled correctly' and 'Try different, more general, or fewer keywords'.

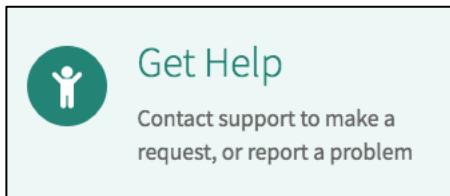
**Note:** Typing either **infinity** or **Infinity** will result in **No Results**.

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.

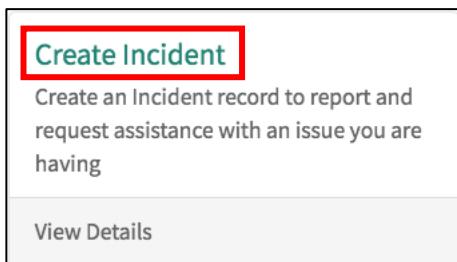
6. Click the **ServiceNow logo** in the top-left corner to return to the homepage.

## B. Submit an Incident

1. Select **Get Help**:



2. Locate and select **Create Incident**.



3. Fill out the form as follows:

**Urgency: 3 – Low**

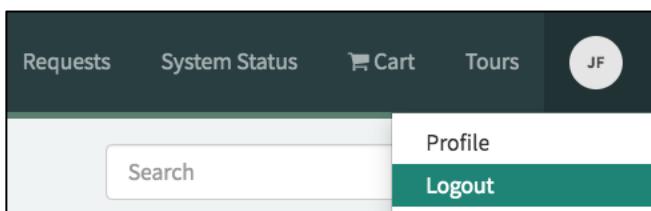
**Please describe your issue below:** *The Infinity is having trouble displaying clearly.*

4. Select **Submit**.

A screenshot of the 'Create Incident' form. The form has several fields:

- A title 'Create Incident' at the top.
- A descriptive text field: 'Create an Incident record to report and request assistance with an issue you are having'.
- A main text area: 'Request assistance with an issue you are having. An incident record will be created and managed through to successful resolution. You will also be notified of progress.'
- An 'Urgency' dropdown menu set to '3 - Low'.
- A text area for 'Please describe your issue below': 'The Infinity is having trouble displaying clearly'.
- A large teal 'Submit' button on the right side of the form.

5. Log out of the instance as **Jon Floyd** using the User menu



**Note:** If you opted to use another browser for these steps, return to the original session and move to step 1 of the Update the Incident section below. Otherwise, continue on to the next step.

6. From your instance URL, remove the **/sp** suffix (including everything that follows).
7. Press **Enter** on your keyboard to return to the normal login screen.

## C. 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**.

**Note:** The incident was automatically assigned to Rita's group (**Infinity Customer Support**) by an existing assignment rule created for the purpose of this exercise.

4. Locate and *open* the most recently created incident record and update the following:

**State:** *In Progress*

**Assigned to:** *Rita Center*

Number	INC0010003	Contact type	Self-service
* Caller	Jon Floyd	State	In Progress
Category	Inquiry / Help	Impact	3 - Low
Subcategory	-- None --	Urgency	3 - Low
Service		Priority	5 - Planning
Configuration item		Assignment group	Infinity Customer Support
Location		Assigned to	Rita Center

5. **Save** the record using the form context menu.

**Note:** Assume Rita has remotely accessed Jon's device and found no apparent issues.



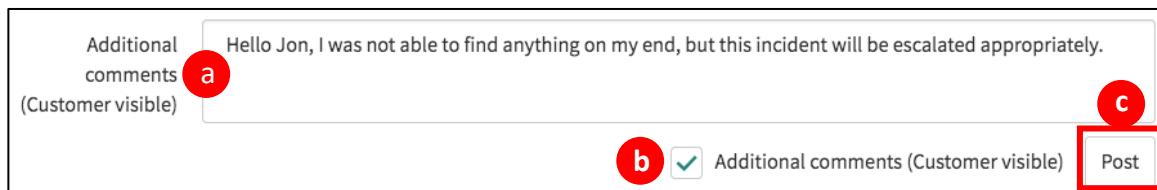
6. Update the incident record to inform Jon that it must be escalated to another team:

a) From the Notes tab, type into the **Work notes** field: **Hello Jon, I was not able to find anything on my end, but this incident will be escalated appropriately.**

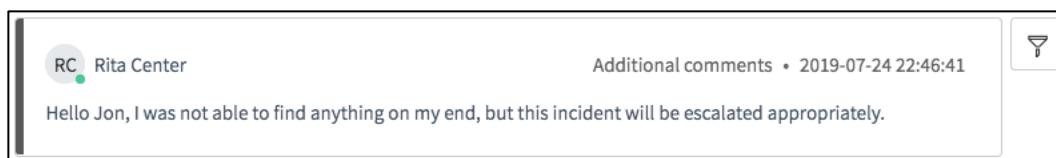
b) Check the **Additional comments (Customer visible)** checkbox

**Note:** Confirm the **Work notes** field label changed to **Additional comments (Customer visible)** before posting.

c) Click the **Post** button



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



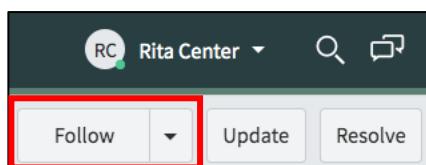
8. Update the incident as follows:

**Category:** *Hardware*

**Assigned to:** *Trey Tout*

9. **Save** (do not update) the record.

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



**Note:** 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.

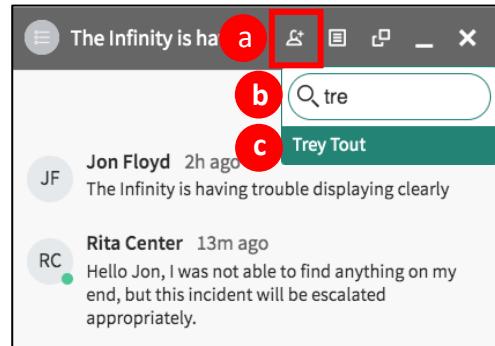
11. Open the **Connect Sidebar** to locate the conversation around the incident:



12. Select the “**Infinity is having trouble**” conversation from the **Connect Sidebar** to open a chat window.

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

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



14. With Trey added to the conversation, type the following message into the **Worknote** 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!*

15. Press **Enter** on your keyboard to save the message.

## D. 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 System Administrator before impersonating again.*

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



3. Click on the conversation to view the chat history.

4. Open the incident by selecting the **View Document** icon on the chat window header.



5. Once the record form is visible, close the chat window and collapse the Connect Sidebar.

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

The screenshot shows the 'Work notes' section of the Notes tab. At the top, there are 'Watch list' and 'Work notes list' buttons. Below that is a text input field labeled 'Work notes' containing the text 'Work notes'. At the bottom right of the input field is a red-bordered 'Post' button.

**Note:** The **Show all journal fields** displays both **Work notes** and **Additional comments**, making it easy to add both internal and external comments to the record.

- Update the **Work notes** with the following:

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

- Update the **Additional comments (Customer visible)** field to say the following:

*Hello Jon, I have been added to help resolve your issue. Could you please provide additional information about what you are experiencing? Thank you.*

Your fields should look like this:

The screenshot shows the Notes tab with two fields. The 'Work notes' field contains the text 'Not enough information provided, contacting the customer to request more.'. The 'Additional comments (Customer visible)' field contains the text 'Hello Jon, I have been added to help resolve your issue. Could you please provide additional information about what you are experiencing? Thank you.'

- Click the **Post** button to add the Work notes and Additional comments to the incident activity stream.

- Set the record **State** to **On Hold**.

- Select **Awaiting Caller** as the **On hold reason**.

The screenshot shows the incident details page. It includes fields for Number (INC0010003), Contact type (Self-service), State (On Hold), On hold reason (Awaiting Caller), Category (Hardware), Subcategory (-- None --), and Impact (3 - Low). The 'On hold reason' dropdown is highlighted with a red box.

12. Notice the Connect Sidebar icon has changed, indicating two new messages have been received:



13. 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.

## LAB VERIFICATION

### Updates to an Incident

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

The screenshot shows the Activity feed for an incident. It contains the following messages:

- Trey Tout: Incident state - On Hold was In Progress (Field changes • 2018-05-29 13:49:01)
- Trey Tout: Additional comments • 2018-05-29 13:48:33  
Hello Jon, I have been added to help resolve your issue. Could you please provide additional information about what you are experiencing? Thank you.
- Trey Tout: Work notes • 2018-05-29 13:48:33  
Not enough information provided, contacting the customer to request more.
- Rita Center: Work notes • 2018-05-29 13:45:49  
Hello Trey, I thought you could help with this as there are no software issues detected. Thanks!
- Rita Center: Field changes • 2018-05-29 13:45:22  
Assigned to Trey Tout was Rita Center
- Rita Center: Additional comments • 2018-05-29 13:45:07  
Hello Jon, I was not able to find anything on my end but this incident will be escalated appropriately.
- Rita Center: Field changes • 2018-05-29 13:44:09  
Assigned to Rita Center  
Incident state In Progress was New

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

**You've learned how easy it is to submit an incident from the Service Portal, locate the incident in ServiceNow, and collaborate using Connect Chat to resolve the incident!**

## Knowledge Check

Which module displays a list of tasks assigned to a user's group, but not yet assigned to an individual?

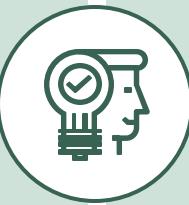
- **My Groups Work**

What is the difference between *Work notes* and *Additional Comments*?

- **Work notes are visible to only those who can view the task**
- **Additional comments are visible to the customer (requestor)**

Which tab contains the activity stream for a task?

- **Notes**



Which tool enables users to collaborate on a task real-time and links conversations to records?

- **Connect Chat**

## Section 2.2: Notifications

### User Story

---

As the **Data Security Manager**,

I want the assignment group manager, and me, to **be notified** whenever a critical (P1) security incident is created

So we can **reduce our exposure** by properly prioritizing resources to resolve the incident quickly.



# Events and Notifications

## Events

An **event** is an indication that certain conditions have occurred in the system, which are then responded to with pre-defined actions

Events are triggered by:

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

## Notifications

Email **notifications** can be triggered by events and require no scripting knowledge

The image shows a sample email notification with a white background and black text. At the top, it says "Preview Email". Below that, in bold, is "Incident has been closed.". Underneath, it says "Summary details" followed by "Closed by: System Administrator" and "Closed notes: Fixed". It then says "You can view all the details of the incident by following the link below:" and provides a blue button with the text "Take me to the Incident". At the bottom, it says "Thank you." and "Ref:MSG000004".

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 Registration** [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**.

A notification is a tool for alerting users when events that concern them have occurred through the following methods:

- Email
- SMS
- Meeting Invitation

Notifications are received by configured users and voluntary recipients and can notify of specific activities in the platform, such as updates to incidents or change requests.

# Creating Notifications

The diagram illustrates the three-step process for creating notifications:

- Choose When to Send**: A screenshot of the "When to send" configuration screen. It shows tabs for "When to send", "Who will receive", and "What it will contain". Under "Send when", "Record inserted or updated" is selected, with "Updated" checked. Below, "Inserted" is checked. Conditions are listed with "Add Filter Condition" and "Add 'OR' Clause" buttons.
- Choose Who will Receive**: A screenshot of the "Who will receive" configuration screen. It shows tabs for "When to send", "Who will receive", and "What it will contain". Under "Users", "Groups", and "Subscribable", checkboxes are shown. Below, "Users/Groups in fields" is checked.
- Choose What it Says**: A screenshot of the "What it will contain" configuration screen. It shows tabs for "When to send", "Who will receive", and "What it will contain". The "Text" tab is active, displaying rich text editor controls (B, I, U, etc.) and a preview area. The text area contains three mail script snippets: `\${mail\_script:incident\_has\_been\_closed}`, `\${mail\_script:incident\_closed\_incident\_details}`, and `\${mail\_script:incident\_take\_me\_to\_the\_incident}`.

**Feature Use Cases**

- HR may use notifications to communicate an office closure due to inclement weather
- Inform customers of the latest information on their open incident
- An account executive is notified that a contract is expiring

Creating notifications allows 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 a table, only if the specified conditions are met.

Send notifications to specific users and/or 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.

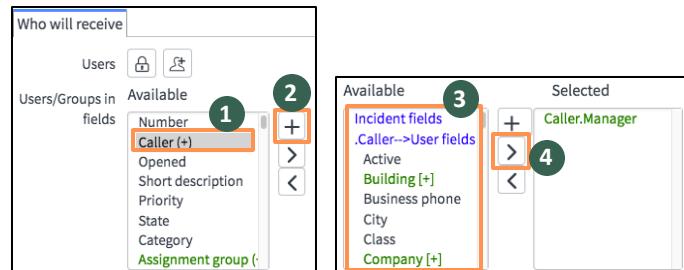
**TIP:** Consider limiting the recipient list of any notification to 1000 users. By default, if a notification has more than 100 intended recipients, the system creates multiple notification messages with up to 100 recipients each. If you want to change the recipient limit, set the system property `glide.email.smtp.max_recipients`.

If using an **Email Template** then **Subject** and **Message** will be used from the template unless overridden with a different (new) subject and/or message.

# Notifications: Dot-Walking

**Dot-Walking** gathers information from a series of tables through reference fields

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 ( > )



Imagine sending a notification when an update is made to an incident.

The recipients of this notification include the user who is experiencing an issue and the user that is assigned to resolve the incident. In the base system, this is easily achievable because the fields representing these users are a part of the incident record by default (**Caller** and **Assigned to**, respectively).

A variation of this could include sending the notification to the caller's manager as well, but this would require dot-walking because the **Manager** field (and value) is not found on the incident table.

To achieve this while defining a notification, dot-walk through the Caller field to the User table. This is possible because the value stored in the Caller field is simply a user record, referencing data stored on the User table. After dot-walking to the User table, all of this table's fields (and their stored data) is accessible, including the Manager field.

In the example illustrated above, the caller of an incident is **David Loo**. David's manager, Bud Richman, is also able to receive the notification because of dot-walking.

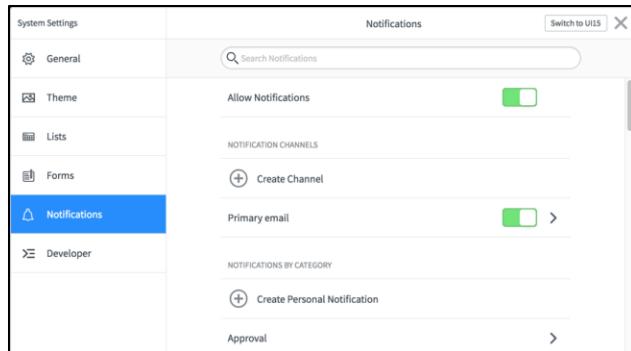
Another benefit to using dot-walking is that notifications do not rely on hardcoded data – instead, the values are automatically populated based on the fields selected. In other words, this notification will automatically identify the caller and the caller's manager (if applicable) based on the information provided within the incident.

## Notifications: Subscriptions

# Subscriptions

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

In addition to email notifications, Short Message Service (SMS) is supported, also known as text messages on mobile devices



The Notifications page of 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 devices. Most mobile phones support SMS, even if they do not support more sophisticated messaging, like 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.

# Connect Chat and Workspace

now.

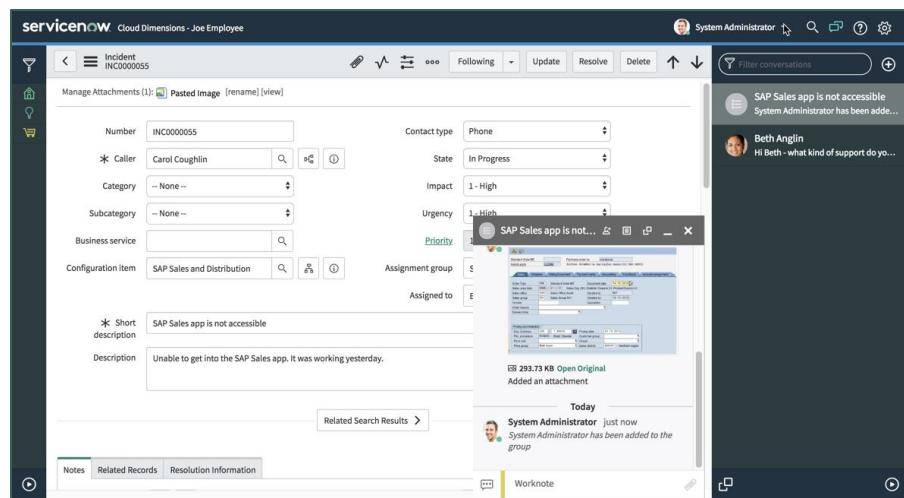
## Chat

**Connect Chat** is a messaging tool that lets you work with others in real-time

## Workspace

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

Click the **Open Connect standalone 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 Connect icon in the Banner Frame (opens a sidebar within the 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 **Connect > 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.



**Time**  
**10-15m**

- Develop a new email notification
- Test the notification

## Lab 2.2: Notifications

As the **Data Security Manager**,

I want the assignment group manager, and me, to **be notified** whenever a critical (P1) security incident is created

So we can **reduce our exposure** by properly prioritizing resources to resolve the incident quickly.

# Notifications

Lab  
2.2

⌚10-15m

## Lab Objectives

You will achieve the following objectives:

- Develop a new email notification
- Test the notification

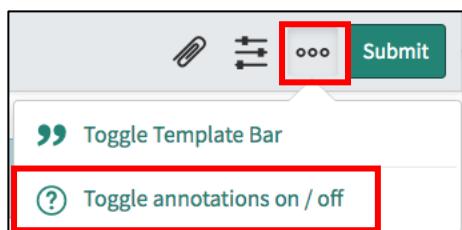
## Scenario

Buster Wubbel – manager of Infinity Security – has requested that a notification be created to alert him and his team whenever a critical employee Infinity incident is active and categorized with Security. Additionally, the notification should alert any manager of the assigned group, seeing as the incident as urgent.

### A. Develop a Notification

We will assume Buster Wubbel has shared the notification requirements with the system administrator – whom you will impersonate to create the new notification.

1. Impersonate **System Administrator**
2. Navigate to **System Notification > Email > Notifications**
3. **Select New**
4. After the new the **More options** menu, 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 Incident**.

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

**Note:** This notification will be defined to inform the Infinity Security group and the assignment group's manager whenever a critical (Priority 1) employee Infinity security incident is active as a result of being created or updated.

6. Select **Incident [incident]** from the **Table** field.

Once the table has been identified, you will configure When to send the notification, Who will receive the notification, and What the notification will contain.

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

Inserted: **[checked]**

Updated: **[checked]**

Conditions:

Active | is | true AND

Priority | is | 1 – Critical AND

Category | is | Security AND

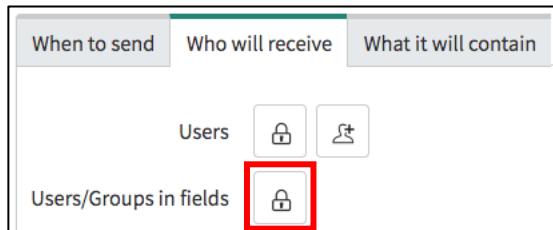
Short description | contains | infinity

OR Short description | contains | Infinity

When to send	Who will receive	What it will contain																				
Send when Record inserted or updated <input checked="" type="checkbox"/> Inserted	Updated <input checked="" type="checkbox"/>																					
Conditions <input type="button" value="Add Filter Condition"/> <input clause"="" or"="" type="button" value="Add "/>	All of these conditions must be met																					
<table border="1"><tr><td>Active</td><td>is</td><td>true</td><td>AND</td></tr><tr><td>Priority</td><td>is</td><td>1 - Critical</td><td>OR</td></tr><tr><td>Category</td><td>is</td><td>Security</td><td>X</td></tr><tr><td>Short description</td><td>contains</td><td>Infinity</td><td>AND</td></tr><tr><td>or</td><td>Short description</td><td>contains</td><td>infinity</td></tr></table>			Active	is	true	AND	Priority	is	1 - Critical	OR	Category	is	Security	X	Short description	contains	Infinity	AND	or	Short description	contains	infinity
Active	is	true	AND																			
Priority	is	1 - Critical	OR																			
Category	is	Security	X																			
Short description	contains	Infinity	AND																			
or	Short description	contains	infinity																			

**Note:** The **Security** category choice was previously defined for this exercise.

- Select the **Who will receive** tab.
- Click the **Unlock** icon (closed lock) for **Users/Groups in fields**:



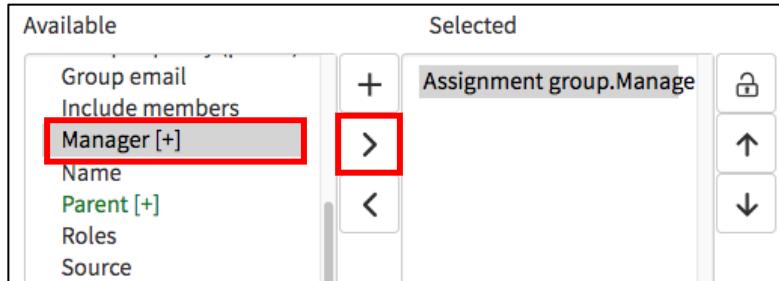
**Note:** Instead of selecting a user by name and hardcoding a specific user to the notification, select a field that contains the data of a user account. Doing so will require dot-walking tables.

- Select **Assignment group (+)** under the **Available** bucket and click the **(+)** to expand.



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

- Scroll down and select **Manager [+]** under **Assignment group**
- Select the **Add Item** icon (**>**).



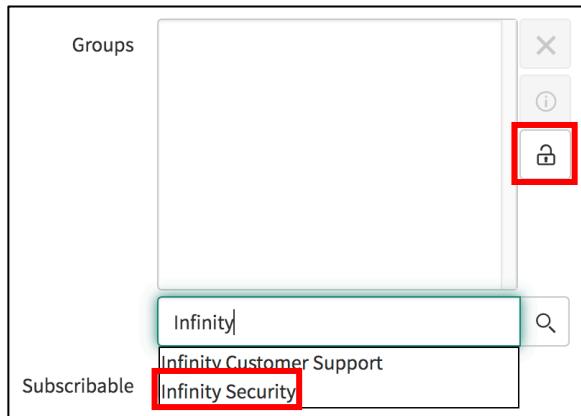
**Note:** This adds **Assignment group.Manager** to the **Selected** bucket or, in other words, the notification will be sent to the manager of the incident's assignment group.

- Click the **Edit Groups** icon (closed lock) on the **Groups** field:



14. Type **Infinity** in the Search box

15. Select **Infinity Security**.



16. Select the **Lock Groups** button (open lock) to lock the selection

Now that you have configured *who* will receive the email notification, it's time to configure *what* the email notification will contain.

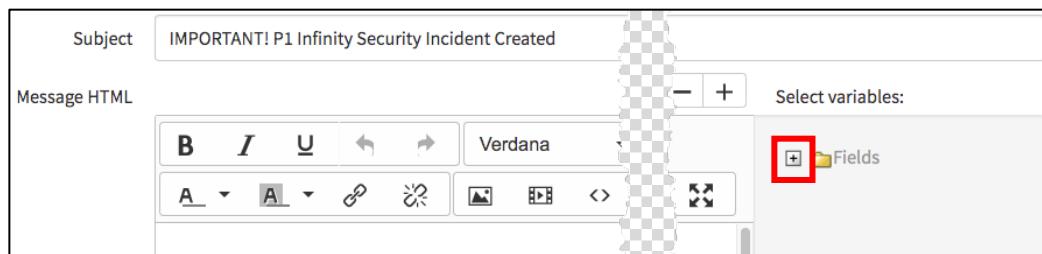
17. Select the **What it will contain** tab.

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

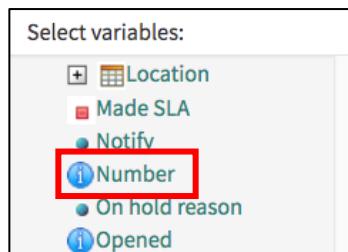
*IMPORTANT! P1 Infinity Security Incident Created*

19. Add a dynamic value placeholder to the subject:

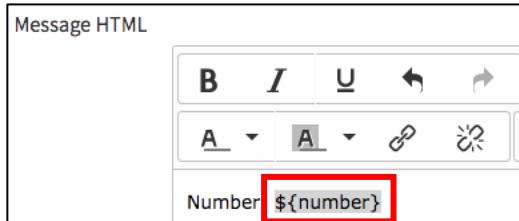
a) From **Select variables**, expand the **Fields** list by clicking the + icon



b) Scroll down and select the **Number** field variable



- c) Highlight \${number} in the **Message HTML** field and copy the text to your clipboard



- d) Place your cursor at the end of the text in the **Subject** field  
e) Press the **spacebar** on your keyboard once  
f) Paste the \${number} text from your clipboard

Subject	IMPORTANT! P1 Infinity Security Incident Created	\${number}
---------	--	------------

**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.

20. 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.

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

## B. 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 **Buster Wubble**.
2. **Incident > Create New**.
3. Fill out the record as follows:

**Caller: Buster Wubble**

**Category: Security**

**Impact: 1 - High**  
**Urgency: 1 - High**  
**Priority: 1 – Critical** (autofills)  
**Assignment group: Service Desk**  
**Short description: Testing P1 Infinity Security Notification**

Number	INC0010004	Contact type	-- None --
* Caller	Buster Wubbel	State	New
Category	Security	Impact	1 - High
Subcategory	-- None --	Urgency	1 - High
Service		Priority	1 - Critical
Configuration item		Assignment group	Service Desk
Location	Karmelitska 2, Lesser Town, P	Assigned to	
* Short description	Testing P1 Infinity Security Notification		

4. Submit the incident.

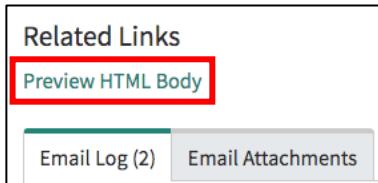
## LAB VERIFICATION

*Email is disabled on the student instance, so you will need to impersonate the System Administrator to check the instance email logs.*

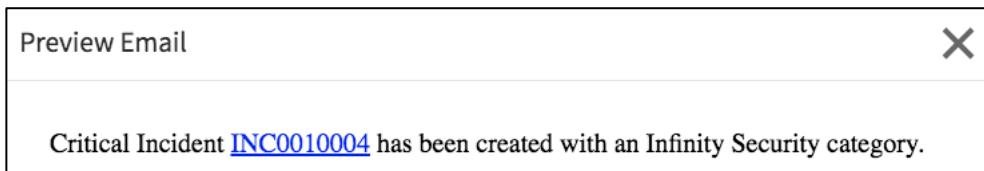
1. Impersonate **System Administrator**.
2. **System Mailboxes > Outbound > Outbox**
3. Locate the record with the Subject **IMPORTANT! P1 Infinity Security Incident Created INC#####**, then click on the **Created** timestamp to open the record.

Emails [Outbox view]				<b>New</b>	Search	Created	Search	◀◀	◀	▶	▶▶	1
<b>All &gt; Mailbox = Outbox &gt; Created on Today</b>												
<input type="checkbox"/>	<input type="radio"/>	<b>Created</b>	▼	<input type="checkbox"/>	Recipients	<input type="checkbox"/>	Subject					
		Search			Search		Search					
		2019-			cdisecurity@cloudd.com,winnie.reich@clou...		IMPORTANT! P1 Infinity Security Incident Created - INC0010004					

4. Scroll down to the **Related Links** section of the form and select **Preview HTML Body** to display a preview of the message received by the recipients:



5. Verify the correct incident number is displayed and that it links to the incident form.



**Great work! You have successfully created and tested a new email notification.**

## Knowledge Check

What can trigger a notification?

- Record inserted or updated
- Event is fired
- A script

What is used to get information from tables through referenced fields?

- Dot-Walking



Which three tabs are used to configure an email notification?

- When to send
- Who will receive
- What it will contain

How is dot-walking used when configuring a notification?

- Identify the recipient (e.g. Users/Groups in fields)
- Populate data in the notification (e.g. \${number} or \${URI\_REF})

## Section 2.3: Reporting

### User Story

---

As the **Service Desk Manager**,

I want a report which provides a **high-level summary of all incidents assigned**; prioritized by category and priority

So I can **understand the incident backlog** and provide additional support where needed.



# What is ServiceNow Reporting?

## Reports



**Reports**, prepared on an ad-hoc basis, show results by allowing users to view and analyze ServiceNow data

Run predefined reports or create new custom reports with the Report Designer

### Real World Use Cases

- Management needs to have a report automatically generated and delivered to her division for all P1 incidents that happened overnight
- Vendor management needs to see a quarterly roll up of Service level compliance of their third party service providers
- HR may use reports to measure average Benefits Case load by assignee during open enrollment
- An agent needs to have evidence of his performance managing cases over the last quarter in preparation for his quarterly review

Reports can be visually represented in many different ways, including bar charts, pie charts, dials, lists, pivot tables, donuts, and more. Reports can be run manually or scheduled to be run automatically. 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.

ServiceNow reports are interactive. Users with access can drill down into the report gauges to view and manipulate the underlying data.

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 context menus in any list to generate a report directly from the data in that list.

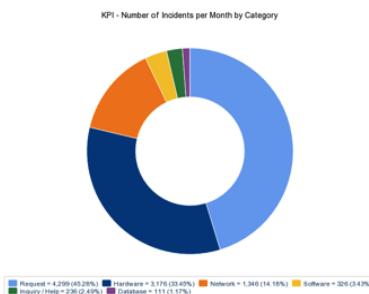
ServiceNow Reporting can easily answer such questions as:

- 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?

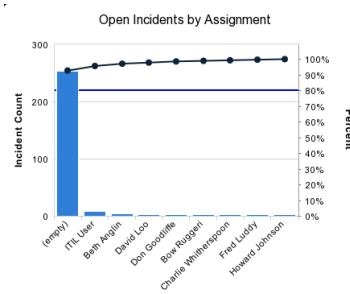
For more information, navigate to **Reports > Getting Started**.

# Report Types

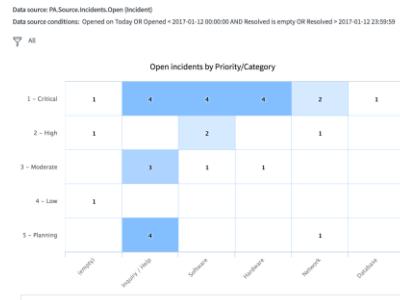
The ServiceNow base instance comes with over 25 standard report types, including:



Donut



Pareto



Heatmap

Other report types include: Speedometer, Dial, Single Score, Pie, Semi Donut, Bubble, Multi-Level Pivot Table, Line, Column, Area, Spline, Bar, Histogram, Horizontal Bar, List, Funnel, Calendar, Pyramid, Box, Trend, Control, Trendbox, Map, Pivot Table, and Text Analytics.

One report variation, 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.

# Create and Edit Reports

## Creating a New Report from a List

1. Define and run a filter, displaying only the data to report on

The screenshot shows the Report Designer interface with a filter configuration. A dropdown menu for 'Priority' is open, listing five levels: 1 - Critical, 2 - High, 3 - Moderate, 4 - Low, and 5 - Info. The 'Run' button at the bottom right of the filter panel is highlighted with a red box.

2. Open the Column Context Menu, then choose Bar Chart or Pie Chart

The screenshot shows a context menu for a column titled 'Priority'. The 'Bar Chart' option is highlighted with a red box.

## Creating a New Report or Edit an Existing Report

Navigate to **Reports > View / Run** to create a new report with the Report Designer or open an existing report

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.

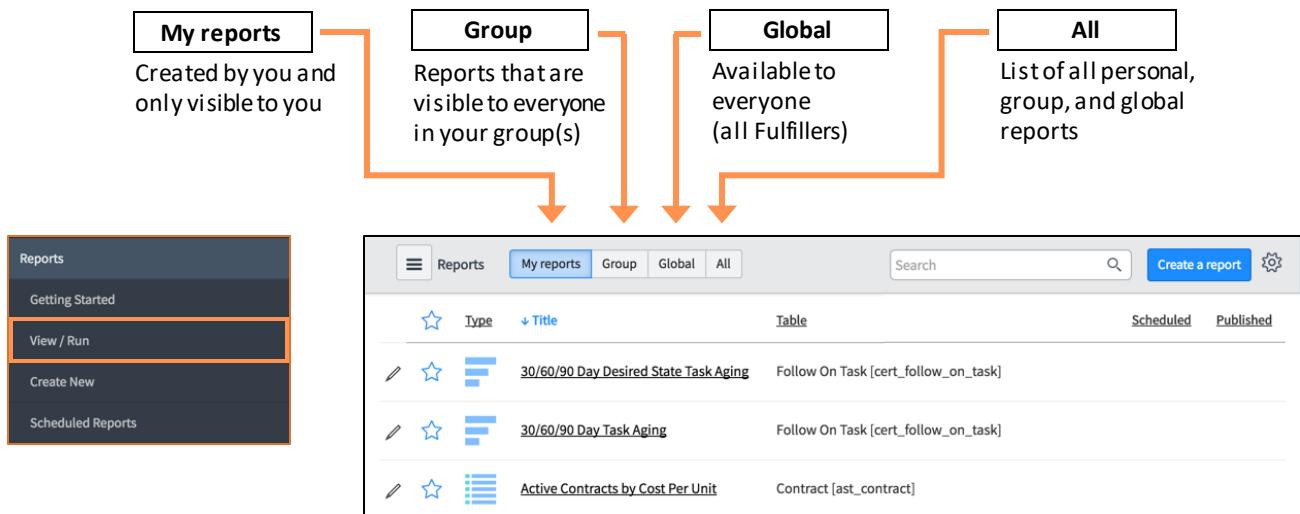
Use the ServiceNow Report Designer to:

- 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 Visibility Controls



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.

# Report Designer: Creating a Report

**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.

The screenshot shows the 'Create a report' interface with the 'Data' tab selected. The form includes fields for 'Report name', 'Source type' (set to 'Data source'), and 'Data source' (showing 'No data source selected'). A 'Next' button is at the bottom. To the right, a large text area says 'Enter the title, select the data you want to visualize and press Next'.

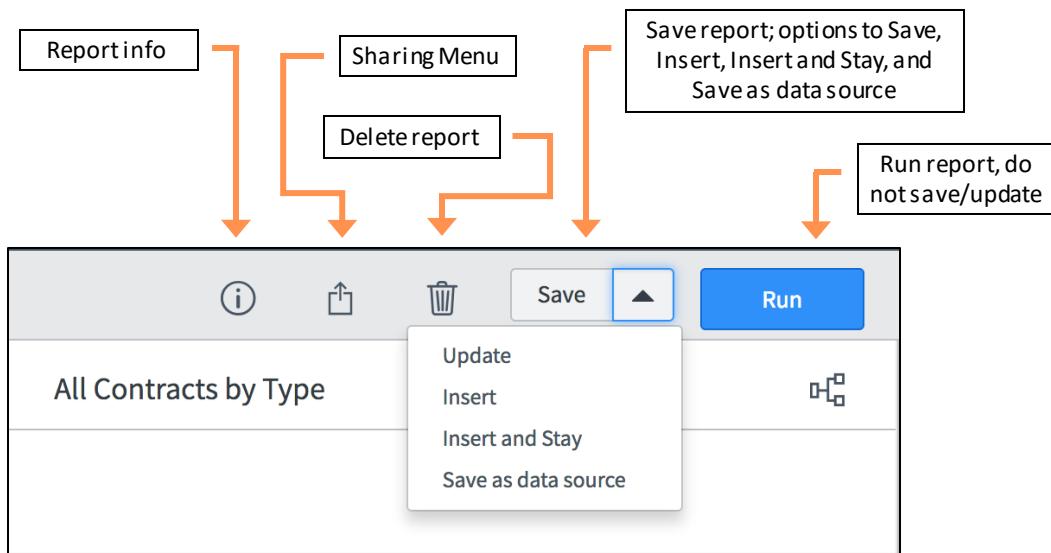
Each section of the Report Designer provides different configuration options:

- **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 28 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

now.



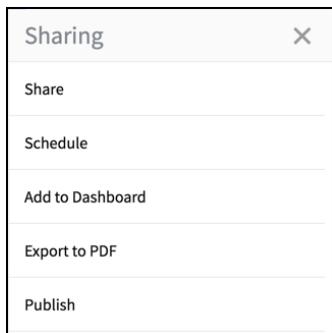
Report actions become available once the report has been saved and they depend on your role. Many of the actions are easily understood; therefore only some of them are detailed here:

- **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

## 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

When distributing a report, sharing has the ability to make the report visible to authenticated users within ServiceNow.

From the **Sharing** menu, the following options are available:

- **Share:** Specifies who can see the report. Options include:
  - Me (the user who created the report)
  - Everyone (all ServiceNow users)
  - Groups and Users (specific groups and/or specific users)
- **Schedule:** Creates a scheduled email of the report as an attachment
- **Add to Dashboard:** Adds directly to a Dashboard on a homepage you choose, or within a Performance Analytics tab
- **Export to PDF:** Converts the report to a PDF which can be generated immediately or sent as an email
- **Publish:** Creates a URL for the report which can be used by internal and external audiences. However, users may need to log into ServiceNow to view all of the data. Published reports can be unpublished and no longer accessible.

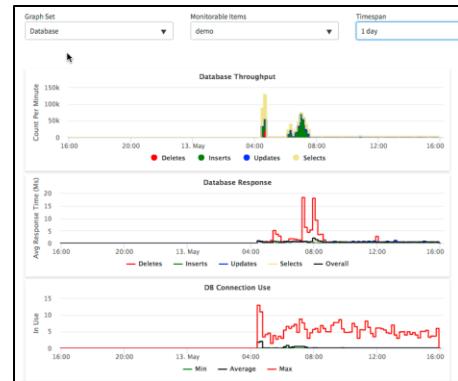
For more information on options available under the sharing menu in Report Designer, please refer to the ServiceNow product documentation by searching for the article titled: **Share a report - Report Designer**.

## Reports and Metrics

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 *measure 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.

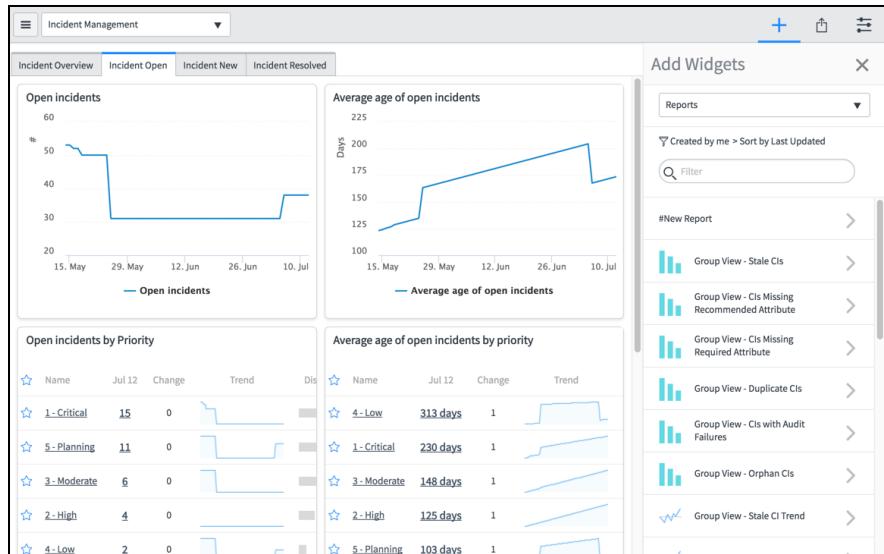
Sometimes a metric can be easily obtained from the data. For example, to find the number of incidents that were created today, a report will count the number of incident records in the Incident table with a created date of today. Other times, metrics need to be gathered as data is updated. For example, 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.

# Dashboards

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.



**Time**  
**10-15m**

- Create a report of assigned incidents
- Share the report to a group

## Lab 2.3: Reporting

As the **Service Desk Manager**,

I want a report which provides a **high-level summary of all incidents assigned**; prioritized by category and priority

So I can **understand the incident backlog** and provide additional support where needed.

# Reporting

Lab  
2.3

⌚10-15m

## Lab Objectives

You will achieve the following objectives:

- Create a report for a group
- Share the report to a group

## Scenario

Members of the Service Desk would like a report built which provides a high-level summary of all incidents assigned to their group, organized by incident category and priority.

Winnie Reich will create a report and then have the system administrator share it to her team.

### A. Create a Report

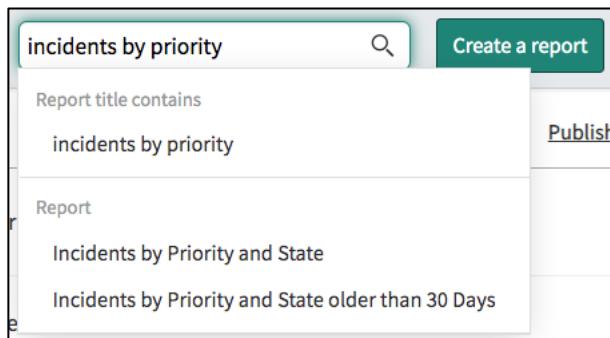
1. Impersonate **Winnie Reich**.
2. Open the list of Reports by navigating to **Reports > View / Run**.

**Note:** Although the first part of the lab is completed as Winnie Reich, any user with the correct role can access the Reports application menu and modules to create and share reports.

3. Click on **All** from the header options.



4. Use the search bar at the top-right to search for reports containing **incidents by priority** in their title.



5. Once the results display, select **Incidents by Priority and State** to open the report:

The screenshot shows the report search results with the "Incidents by Priority and State" report highlighted with a red box. The report title is "Incidents by Priority and State" and it is categorized under "Incident [incident]".

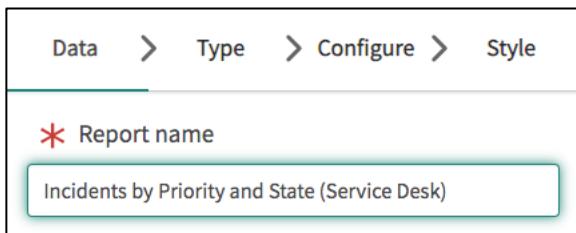
6. The **Report Designer** loads with the report displaying information:

The screenshot shows the Report Designer interface with the report titled "Incidents by Priority and State". The configuration pane on the left shows settings for Row (Priority), Column (State), Aggregation (Count), and Max number of groups (20). The main pane displays a pivot table with the following data:

	1 - Critical	2 - High	3 - Moderate
1 - Critical	3	11	4
2 - High		4	
3 - Moderate	3	3	1

**Note:** Your report may look different than the screenshot provided.

7. Select the **Data** breadcrumb title from the panel on the left, then update the name of the report to **Incidents by Priority and State (Service Desk)**:



8. From the main reporting area, where the data is displayed, click the **Open condition builder** icon (funnel) to apply a filter on the data:

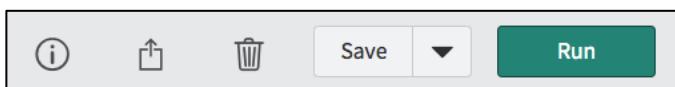
A screenshot of the reporting area titled "Incidents by Priority and State (Service Desk)". It shows the data source as "Incidents.Open (Incident)" and the condition as "Active = true". A red box highlights the "All" filter icon, which is a funnel symbol.

9. Use the drop-down menus to set a single filter condition:

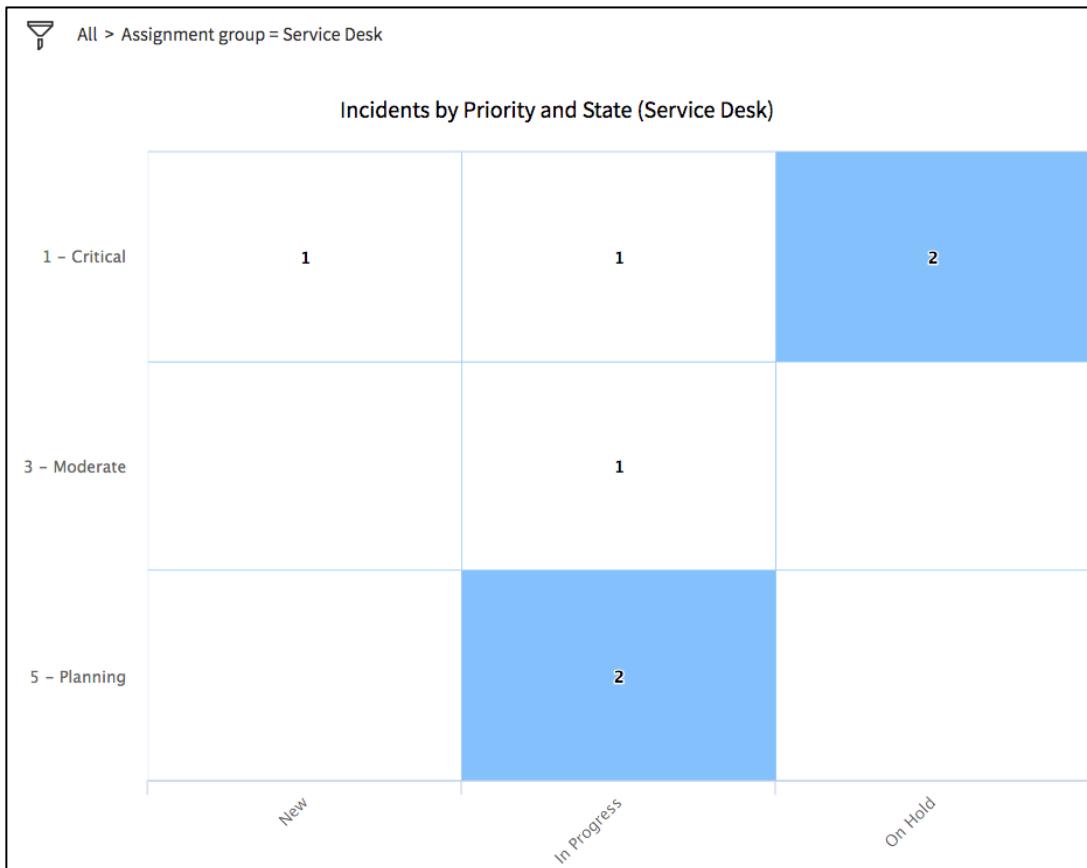
**Assignment group | is | Service Desk**

A screenshot of the "Condition Builder" dialog. It shows a dropdown menu for "Assignment group" with "Service Desk" selected. The condition is defined as "Assignment group is Service Desk".

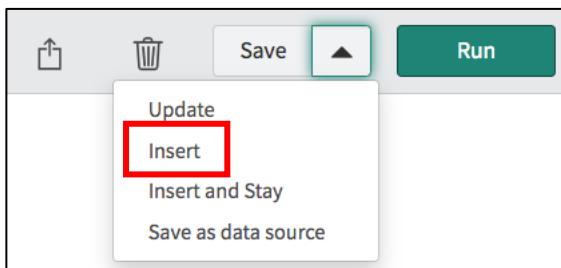
10. From the Report Designer header, click the **Run** button:



11. Notice the report's data has updated:



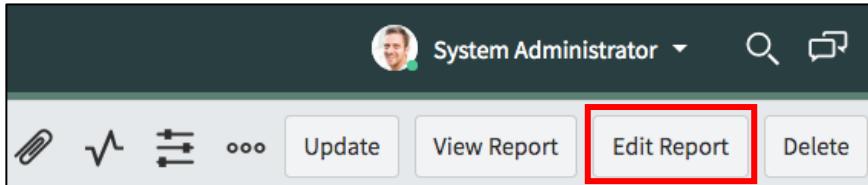
12. Open the **Save** menu by clicking the downward-facing arrow, then select **Insert** to save a copy of the new report you created:



## B. Share the Report

1. Impersonate System Administrator
2. Reports > Administration > All
3. Locate and open the record for the **Incidents by Priority and State (Service Desk)** report.

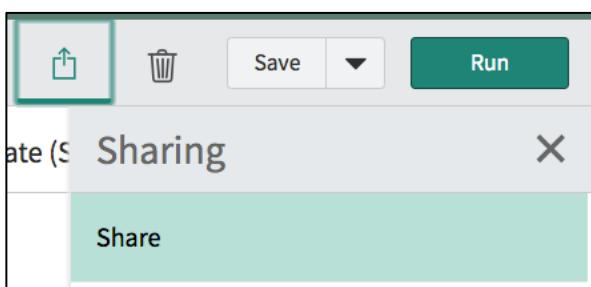
4. Click the **Edit Report** button:



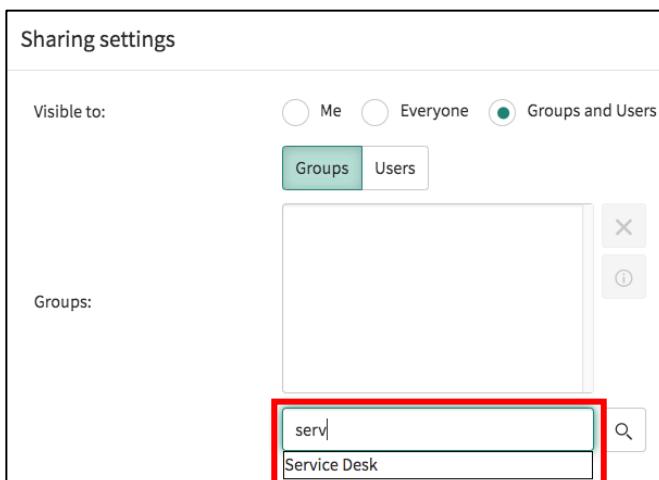
5. From the Report Designer, open the **Sharing** menu:



6. Select **Share**.



7. Choose **Groups and Users**, then add the **Service Desk** to the Groups list:



8. Click **OK** to close the Sharing settings window.

9. **Save** the report.

## Lab Verification

### Access Report from Group Category

1. Impersonate Kevin Edd.
2. **Reports > View / Run**
3. From the Reports list, select the **Group** category:

The screenshot shows the ServiceNow Reports list. At the top, there is a navigation bar with tabs: Reports, My reports, Group (which is highlighted in green), Global, and All. Below the navigation bar is a search bar labeled "Search". The main area displays a table with columns: Type, Title, Table, Created by, and Last modified. There are two rows in the table. The first row has a pencil icon and a star icon next to it. The second row has a star icon and a grid icon next to it. The title for both rows is "Incidents by Priority and State (Service Desk)". The table indicates that the report is a "Table" type, created by "Incident [incident]", and last modified in 2019.

4. Select the **Incidents by Priority and State (Service Desk)** report to verify you (as Kevin) have access to run the report.

The screenshot shows the details page for the "Incidents by Priority and State (Service Desk)" report. At the top, there is a header with a user profile picture for "KE Kevin Edd", a search bar, and a "Run" button. Below the header is a toolbar with icons for information, export, and dropdown menus. The main content area contains the report title "Incidents by Priority and State (Service Desk)". A red box highlights the "Run" button.

**Challenge:** (Optional) Export the report to a PDF and send as an email to yourself

**Wonderful! In this lab you have learned how to modify an existing report and share it with members of a group – well done!**

## Knowledge Check

Which module opens the Report Designer interface?

- Reports > View / Run

Name three ways of sharing a report with other people in the organization

- Scheduled email
- Export to PDF
- Publish

What are the four tabs used to guide the user through creating a new report in the report designer?

- Data, Type, Configure, and Style



What should you consider when exporting a report to PDF and sharing the exported file?

- Information may be obsolete
- Access control is not enforced on exported data

## Module 3: Database Administration

User Interface and Navigation

Collaboration

**Database Administration**

Self-Service and Automation

Introduction to Development

Capstone Project

### Module Objectives

- Explain the hierarchy of tables, records, and fields
- Differentiate the classes of tables in ServiceNow
- Identify different field types in a record
- Demonstrate how to create access control rules
- Discuss how access controls are evaluated
- Explain how to use import sets to populate data in a table
- Describe the purpose of the CMDB
- Create a new CI class in the CMDB along with relationships to other CIs

### Labs and Activities

- Lab 3.1: Create an Extended Table
- Lab 3.2: Application and Access Controls
- Lab 3.3: Import Sets
- Lab 3.4: CMDB

## Section 3.1: Data Schema

### User Story

As the **Infinity Testing Coordinator**,

I want to be able to **store data for tracking** the assignment of Infinity devices in ServiceNow

So I can **effectively support** the Infinity beta testing efforts.



# What is the ServiceNow Infrastructure?

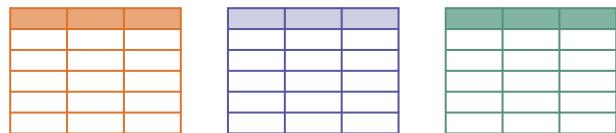
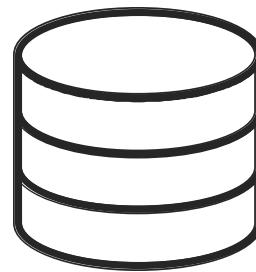
now.

Everything in ServiceNow is built on a relational database containing data which can be accessed through the Now Platform® User Interface.

The ServiceNow Infrastructure includes tables, records, and fields.

Tables contain records. Records correspond to rows in a table.

A field is an individual piece of data in a record and corresponds to a column in the table.



Data in ServiceNow is stored and managed according to a database structure that administrators can view and configure:

- Tables are a data structure or database component, which contain records
- Records are the data stored on tables, which contain fields
- Fields are individual pieces of data within a record

The **System Dictionary** contains the definition for each and every table and field in the database. Navigate to **System Definition > Dictionary** to access the system dictionary to modify table and field attributes.

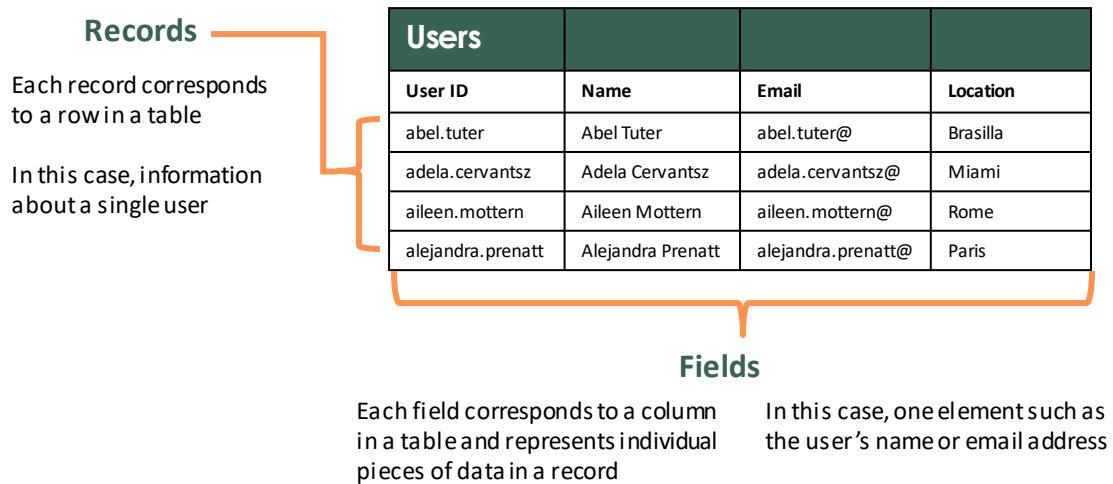
Tables can be accessed using the following modules within the System Definition application:

- **Dictionary** - defines every table and field in the system. Table records are identified as a Collection type.
- **Tables & Columns** - lists existing tables in the database. Selecting a table name will display its contents.
- **Tables** - contains a record for each table in the database. Custom tables can be created when the New button is selected.

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

# Table Components

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 **System Definition > Number Maintenance** application.

For example, the default problem record number prefix is “PRB” but can be redefined as “PRBLM.”

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.

In addition to the System Dictionary, use the **System Definition > Tables & Columns** module to view the field settings and attributes for a table.

# Field Attributes

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

Users		
User ID	Name	Location
abel.tuter	Abel Tuter	Brasilla
adela.cervantsz	Adela Cervantsz	Miami
aileen.mottern	Aileen Mottern	Rome
alejandra.prenatt	Alejandra Prenatt	Paris
alejandro.mascall	Alejandro Mascall	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

## Field Name

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

## Value

The values are actual data

Examples of a field **label** seen above includes User ID, Full Name, and Location

The **name** is a unique term that does not always match a field's label.

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.

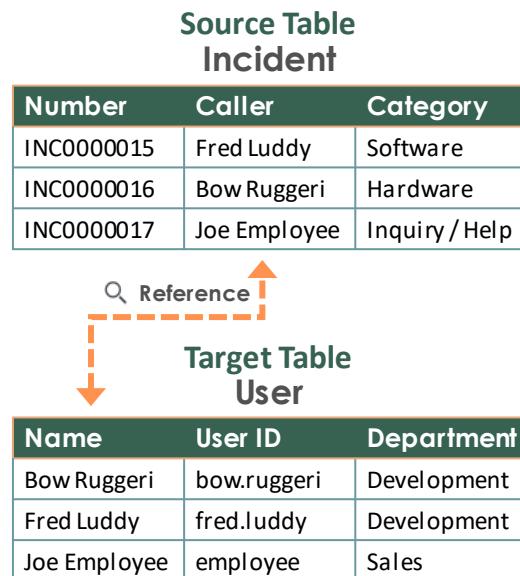
**Values** are the actual data, such as this user's name, Aileen Mottern, or her location, Rome. In some cases, the value may be empty, or null.

## Reference Fields

**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)



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.

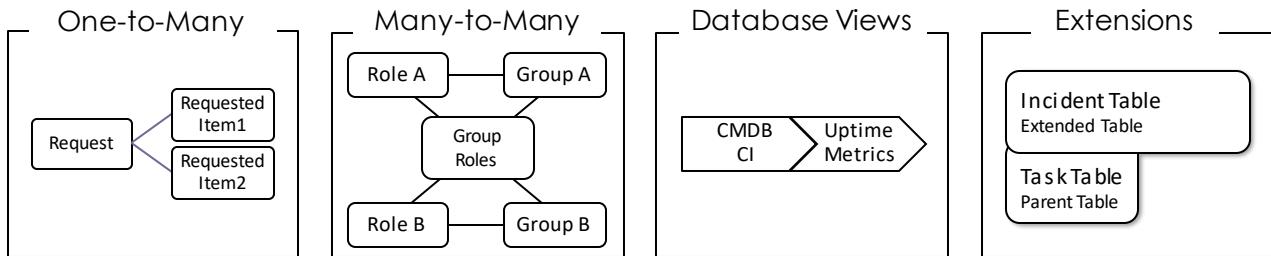
Administrators can create new reference fields and configure several options for reference fields.

**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.

# Table Relationships

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.

# Table Relationships: Extended Tables

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**

**Parent Class**

Task	
Number	Location

A child table inherits the fields of its parent

**Child Class**

Incident		
Number	Location	Caller

A child table may also contain fields that are unique to that child 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.

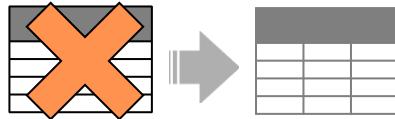
For example, child tables extended from the Task table include Change Request, Incident, and Problem. Child tables extended from the Configuration table include Database, Hardware, and Software.

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.

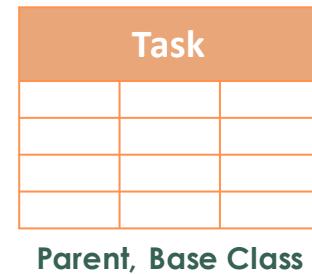
## Table Types: Base Tables

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

Children Classes		
Incident	Problem	Change Request



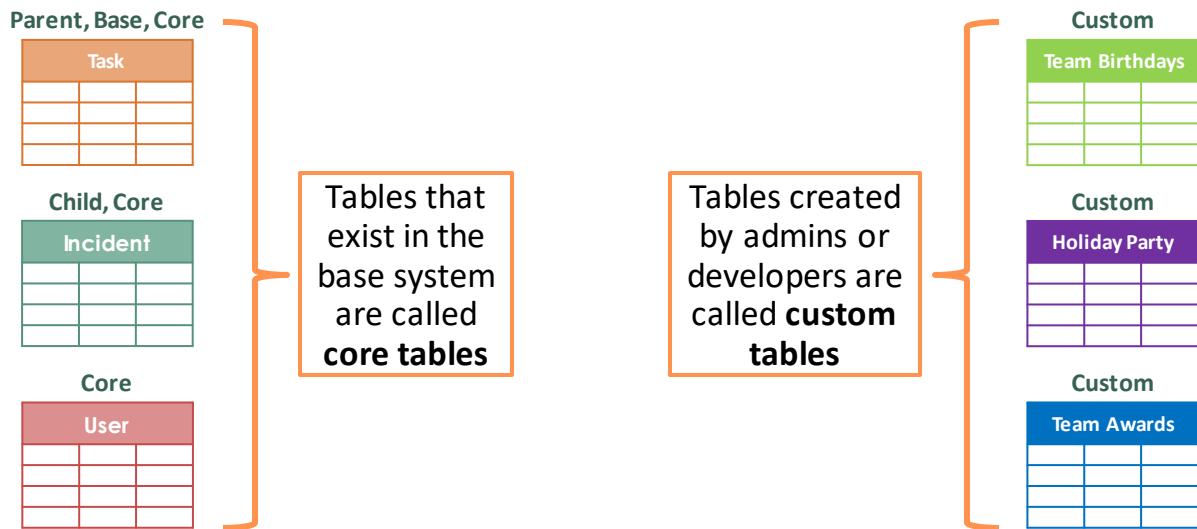
If a field is on a base parent table such as the **Task [task]** table, for example, a different label can be defined for each extended table, such as Incident or Problem. To add a different label for an extended table, navigate to **System Definition > Language File**, then create a new entry for the 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.

## Table Types: Core vs. Custom



Although custom tables are not in the base system, they can still interact with existing core tables.

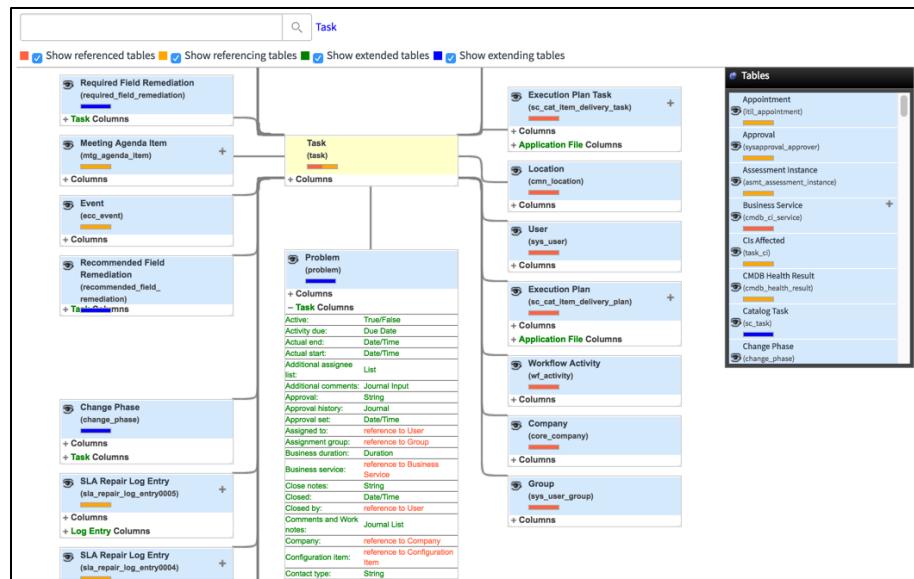
For example, a reference field on a custom table can access data stored on a core table. By doing so, a relationship between the tables is created which makes them related tables. This relationship is not exclusive between just a custom table and a core table. Related tables can be a combination of multiple core tables and/or multiple custom tables.

**NOTE:** When creating a new custom table, the table name is automatically populated based on the table label and a prefix. If the table is being created in a scoped application, the name is prefixed with a namespace identifier, indicating that it is a part of an application. Otherwise, custom tables in the global application feature “`u_`” as their prefix, and then the table label.

# Schema Map

The **Schema Map** provides a graphical representation of other tables related to a specific table.

Relationships can be filtered by extension or reference classes by checking the appropriate boxes at the top of the map.



In this example map, you will see the Task table as the focus of the map (highlighted in yellow).

**NOTE:** The schema map is available to those users with the **personalize\_dictionary** role as well as those with the **admin** role.

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 which 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.



**Time**  
**15-20m**

- Create a new table
- Configure the table form view
- Update the application menu and create new modules

## Lab 3.1: Create an Extended Table

As the **Infinity Testing Coordinator**,

I want to be able to **store data for tracking** the assignment of Infinity devices in ServiceNow

So I can **effectively support** the Infinity beta testing efforts.

# Create an Extended Table

**Lab  
3.1**

⌚15-20m

## Lab Objectives

You will achieve the following objectives:

- Create a new table
- Configure the table form view
- Update the application menu and create new modules

## Scenario

With an active procedure of testing Infinity devices, Cloud Dimensions needs a method for managing inventory and tracking how many devices have been issued and to whom.

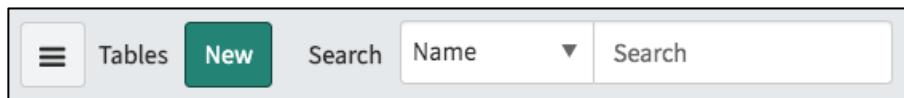
The various teams involved with Infinity testing have come up with a solution but will need the help of the system administrator to implement it.

Their plan is to have this information accessed through an application menu with a series of modules.

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.

### A. Create a New Table

1. Navigate to **System Definition > Tables** to create a new table.
2. From the list header, click the **New** button.



- Fill out the top of the Table form with the following information:

**Label:** *Infinity*

**Name:** *u\_cmdb\_ci\_hardware\_infinity* (auto fills with *u\_infinity*)

**Extends table:** *Hardware [cmdb\_ci\_hardware]*

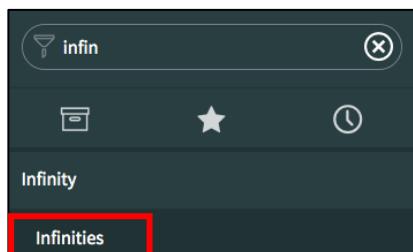
**New menu name:** *Infinity* (auto fills)

**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.

- Submit.

## B. Add Fields to the Infinity Form

- Use the Application Navigator filter field to navigate to **Infinity > Infinities**



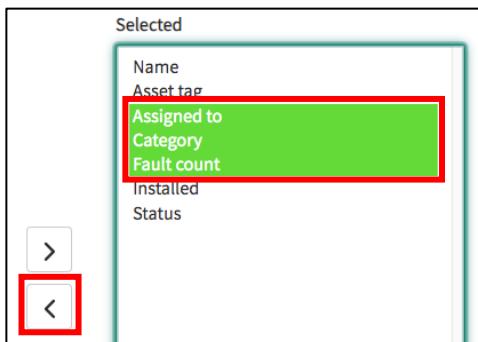
**Note:** During the creation of the *Infinity* table, the **Create module** checkbox was selected.

As a result, a new module, **Infinities** (or **Infinity s** - a pluralized form of the table name) was created.

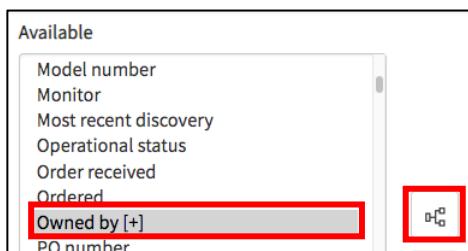
A list with *No records to display* is shown with default fields.

**Note:** In a future lab, you will modify the *Infinities* list layout before importing device records from spreadsheets. For this requirement, we will modify the form layout.

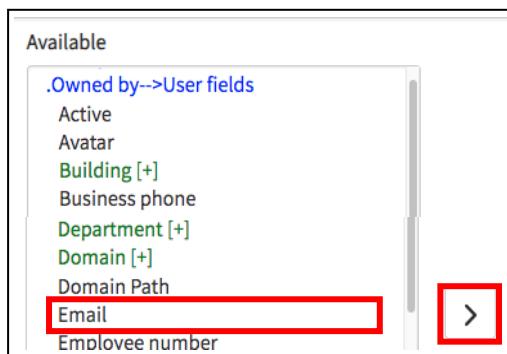
2. Click **New** to open a form displaying default fields.
3. Open the **Form Context Menu**, select **Configure > Form Layout**
4. Remove the **Assigned to**, **Category** and **Fault count** fields, keeping **Name**, **Asset tag**, **Installed**, and **Status** in the Selected list.



5. Add the **Owned by** and **Support group** fields to the Selected list
6. 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) Locate the **.Owned by-->User fields** and add **Email** to the **Selected** list



You should now see Owned by.Email under the Selected list:

The screenshot shows a list titled "Selected" containing several items: Name, Asset tag, Installed, Status, Owned by, Support group, and Owned by.Email. The "Owned by.Email" item is highlighted with a green background.

**Note:** The **Owned by.Email** 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, as long as the user record includes an email address.

7. Next, in the **Create new field** section, enter:

Name: **Device Number**

Type: **String** (autofills)

Field length: **Small (40)** (autofills)

8. Select **Add**.

The screenshot shows the "Create new field" dialog box. It has three input fields: "Name" (Device Number), "Type" (String), and "Field length" (Small (40)). Below these fields is a red-bordered "Add" button.

9. Add two additional fields.

Name	Type	Field Length/Reference
Device Version	String	Small (40)
Device Notes	Journal	

10. After adding the new fields, rearrange the fields under the **Selected** list to look like the image to the right:

11. Select the **Save** button.

12. Close the **Saving Form Section** window.

The screenshot shows the "Saving Form Section" window. It displays a progress bar at 100% complete and the message "Succeeded 100%". At the bottom right is a red-bordered "Close" button.

The screenshot shows the "Selected" list with the following order: Device Number, Name, Asset tag, Device Version, Support group, Installed, Status, Owned by, Owned by.Email, and Device Notes.

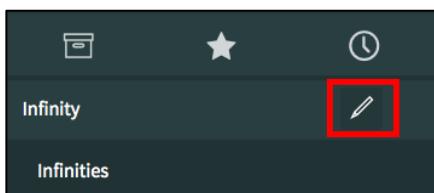
13. The **Infinity** New record form displays:

The screenshot shows the 'Infinity New record' form. It includes fields for Device Number, Name, Asset tag, Device Version, Support group, Installed, Status, Owned by, Email, and Device Notes. The 'Status' field is set to 'Installed'. The 'Owned by' and 'Email' fields have search and envelope icons respectively. The 'Device Notes' field is empty. The top right of the form has a pencil icon, a refresh icon, and a 'Submit' button.

## C. Update the Infinity Application Menu

Now that the Infinity form layout has been configured, it's a good time to modify the application menu name (**Infinity**) as well as the existing module name (**Infinities**) to more user-friendly names.

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.

The screenshot shows the 'Application Menu' record for 'Infinity'. The title is 'Infinity Inventory'. The right side has a context menu with 'Save' (highlighted in green), 'Insert', 'Insert and Stay', and 'Show File Properties'.

- From the Modules section, open the **Infinities** record:

The screenshot shows the ServiceNow Modules list interface. At the top, there are buttons for 'Modules' (disabled), 'New' (highlighted in green), 'Search', 'Order' (with a dropdown arrow), and another 'Search' button. Below this is a toolbar with icons for application menu, title, table, and active status. The main list area shows a single record for 'Infinities'. The 'Title' field contains 'Infinity [u\_cmdb\_ci\_hardware\_infinity]' and the 'Active' field is set to 'true'. The 'Infinities' row is highlighted with a red box around the 'i' icon.

- Update the record as follows:

**Title: All Devices**

**Order: 200**

The screenshot shows the 'Infinities' module edit form. It has fields for 'Title' (set to 'All Devices'), 'Application menu' (set to 'Infinity Inventory'), and 'Order' (set to '200'). The 'Title' and 'Order' fields are highlighted with red boxes.

**Note:** Notice the Role, *u\_infinity\_user* shown under the Visibility tab. This role was created automatically when the Infinity table was created.

In a future lab, we will see how this role is used to manage visibility of the Infinity Devices application and its modules.

- Click **Update**.

## D. Configure a new Infinity Module

The application menu (*Infinity Inventory*) and the module which displays the list of Infinity devices (*All Devices*) is now visible in the application navigator. Now, a new module will need to be configured so users can create new Infinity device records (using the form you have just designed).

- Select **New** from the Modules section.

The screenshot shows the ServiceNow Modules list interface. The 'New' button in the top bar is highlighted with a red box. The main list area shows a new module record for 'All Devices'. The 'Title' field contains 'Infinity [u\_cmdb\_ci\_hardware\_infinity]' and the 'Active' field is set to 'true'.

2. Fill out the form using the following information:

**Title: Add Inventory**

**Order: 100**

3. Click the **Link Type\*** tab and fill out the fields as shown:

**Link type: New Record**

**Table: Infinity [u\_cmdb\_ci\_hardware\_infinity]**

This screenshot shows the 'Link Type' configuration interface. At the top, there are two tabs: 'Visibility' and 'Link Type'. The 'Link Type' tab is currently selected. Below the tabs, there are two input fields. The first field is labeled 'Link type' and contains the value 'New Record'. The second field is labeled '\* Table' and contains the value 'Infinity [u\_cmdb\_ci\_hardware\_infinity]'. Both fields have dropdown arrows to their right.

4. Verify the Title, Order, Link type, and Table are complete, as shown:

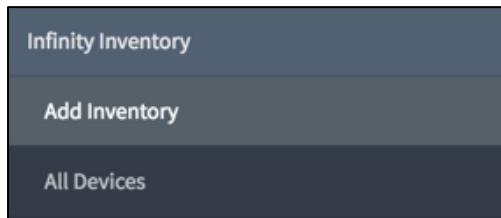
This screenshot shows the 'New record' form for 'Add Inventory'. At the top, it says 'Module: New record' and has a 'Submit' button. Below that, there are several input fields: 'Title' (containing 'Add Inventory'), 'Application menu' (containing 'Infinity Inventory'), 'Order' (containing '100'), 'Hint' (empty), 'Display name' (empty), and 'Visibility' and 'Link Type' tabs. The 'Link Type' tab is selected, showing 'New Record' under 'Link type' and 'Infinity [u\_cmdb\_ci\_hardware\_infinity]' under '\* Table'. The 'Title' and 'Order' fields are highlighted with red boxes, and the entire 'Link Type' configuration section is also highlighted with a red box.

5. **Submit.**

**Note:** 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?

## Lab Verification

1. Verify the application menu, Infinity Inventory, and associated modules are visible in the application navigator.



2. Confirm the Add Inventory form has the correct fields in the correct order.

A screenshot of the "Add Inventory" form. The title bar says "Infinity New record". The form contains the following fields:

- Device Number
- Name
- Asset tag
- Device Version
- Support group
- Installed
- Status (dropdown menu showing "Installed")
- Owned by
- Email
- Device Notes

Each field has a corresponding input box or dropdown menu. At the top right of the form are buttons for "Submit", "Cancel", and "Help".

***Congratulations on completing the lab!***

***Thanks to you, Cloud Dimensions has its first custom table  
that can be used to manage Infinity inventory!***

## Knowledge Check

Which modules can you use to create a new table?

- **Tables**
- **Tables & Columns**

What is created by default when you create a new table?

- **Application Menu with the same name as the table Label (e.g. Infinity)**
- **Module with the plural of the table Label (e.g. Infinities)**

How can you edit the title of an application menu?

- **Click the Edit Application (pencil) icon**
- **Update the Title field on the form**

Which tools can be used to create or modify a form layout?

- **Configure > Form Layout or**
- **Configure > Form Design**



## Section 3.2: Application/Access Control

### User Story

---

As the **Data Security Manager**,

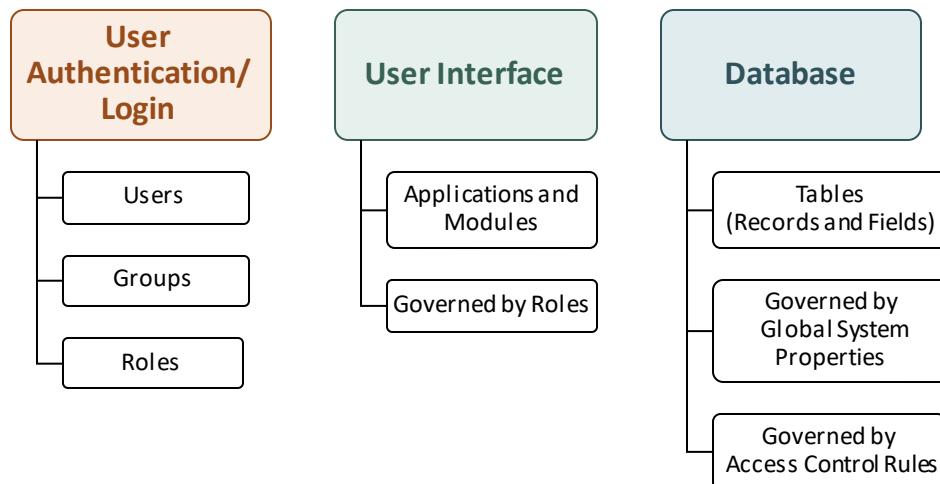
I want to **restrict access** to the Infinity Inventory application menu and modules and **control which fields can be updated**

So I can **ensure the integrity of the inventory data**.



# User Permissions Summary

now.



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 (deny access is the default behavior) as well as table and field level access controls

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?

## Access Control



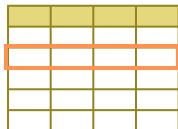
An **Access Control** is a security rule defined to restrict the permissions of a user from viewing and interacting with data

It is executed when attempting to access any ServiceNow table and may be set at the:

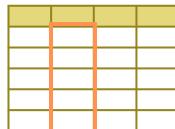
**row-level**

and/or

**column-level**



(access to the record)



(access to the field)

These rules restrict ServiceNow-specific and CRUD operations

**C**reate

**R**ead

**U**pdate (write)

**D**elete

Most security settings are implemented using Access Controls.

In addition to restricting CRUD operations, Access Control rules can restrict 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)

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

Name	Operation	Type	Active	Updated by	Updated
kb_category.create	create	record	true	admin	2014-08-20 07:21:58
kb_category.delete	delete	record	true	admin	2014-08-20 02:36:25
kb_category.read	read	record	true	admin	2017-07-27 01:47:26
kb_category.write	write	record	true	admin	2014-08-20 06:45:38
kb_category.active.write	write	record	true	admin	2014-08-20 07:32:56
kb_category.full_category.write	write	record	true	system	2018-12-19 12:55:20
kb_category.parent_id.write	write	record	true	admin	2018-01-18 01:56:05

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! With this said, in order for a user to create or update Access Control roles, they must have the **security\_admin** role.

# Access Control Definition: Permission Requirements

now

Each Access Control rule specifies

1. The object being secured (e.g. table, field)
2. The permissions required to access the object
  - Roles
  - Conditional Expressions
  - Scripts
3. Operation - a valid action the system can take (CRUD)

The screenshot shows the 'Access Control' interface for defining a permission rule. The top section is labeled 'Access Control change\_request'. It has fields for 'Type' (set to 'record'), 'Operation' (set to 'write'), 'Application' (set to 'Global'), and 'Active' (checkbox checked). Below these are sections for 'Admin overrides' and 'Description' (containing 'itil role required to write to change\_request records'). A large button labeled 'Definition' is present. The 'Definition' section is expanded, showing a table titled 'Requires role' with one row containing the 'itil' role. There is also a link to 'Insert a new row...'.

In this example, a user with the itil role is granted permission to write (update) records in the change\_request table.

Object	Name	Operation	Permissions
table	change_request.None	write	itil role

# System Created Access Controls

now

The screenshot shows the ServiceNow interface for a custom table named "u\_cmdb\_ci\_hardware\_infinity". A callout box labeled 1 highlights the "CONFIG" tab, which is highlighted with an orange border. Another callout box labeled 2 highlights the "Access Controls" tab, also with an orange border. The "Access Controls" tab is active, displaying four default rules:

Name	Operation	Type	Active	Updated by
u_cmdb_ci_hardware_infinity	create	record	true	admin
u_cmdb_ci_hardware_infinity	delete	record	true	admin
u_cmdb_ci_hardware_infinity	read	record	true	admin
u_cmdb_ci_hardware_infinity	write	record	true	admin

With System-created Access Controls, it is rare to have a table with no access control rules. This will be important to remember when we review how these rules are evaluated.

When a table is created, access controls are automatically created. In lab 3.1, the table, u\_cmdb\_ui.hardware\_infinity, was created. In addition to the table, the following were created:

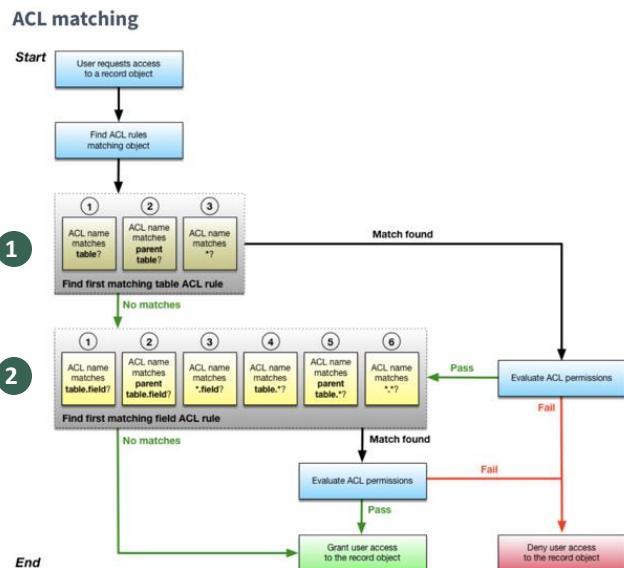
- u\_infinity\_user role
- Table Access Control Rules
  - create
  - delete
  - read
  - write

The screenshot shows the "Access Control" configuration for the "u\_cmdb\_ci\_hardware\_infinity" table. The "Type" is set to "record" and the "Operation" is "create". The "Active" checkbox is checked. The "Admin overrides" dropdown is open, showing "u\_cmdb\_ci\_hardware\_infinity" as the selected name. The "Description" field contains "Default access control on u\_cmdb\_ci\_hardware\_infinity". The "Requires role" section shows the "u\_infinity\_user" role assigned.

Above image shows the default access control on "u\_cmdb\_ci\_hardware\_infinity" granting create, delete, read, and write access to users with the "u\_infinity\_user" role. Users in groups with the "u\_infinity\_user" role will have the ability to create, delete, read, and write (update) the records in the "u\_cmdb\_ci\_hardware\_infinity" table.

# Table Access Control Evaluation

now.



When a session requests data, the system looks for matching access control rules.

Record ACL rules are processed in the following order:

1. Match the object against table ACL rules - *most specific to most general*.
2. Match the object against field ACL rules - *most specific to most general*.

A user must pass **both** table and field ACL rules to access a record object.

Access controls are evaluated in the order shown in the image above. This processing order ensures a user has access to the table (most specific table rules to most general) before evaluating access to the field (most specific to most general).

- If a user fails a table access control rule, the user is denied access to all fields in the table, even if the user would pass a field ACL rule
- If a user passes a table ACL rule, but fails a field ACL rule, the user cannot access the field described by the field ACL rule

If there is a matching access control rule (which is almost guaranteed by system-created access controls), the system evaluates if the user has the permissions required to access the object and operation (role, conditions, script). If an access control rule specifies more than one permission, then the user must meet all permissions to gain access to the object and operation. Failing any one permission check prevents the user from accessing the matching object and operation.

If a user does not meet the permissions of the first matching rule, the system evaluates the permissions of the next matching access control rule as specified by the access control processing order. If the user fails to meet the permissions of any matching access control rule, the system denies access to the requested object and operation.

# Access Control Definition: Rule Types

now.

## table.None

\* Name ▾  
Incident [incident] ▾  
-- None -- ▾

No specific field selected - this rule applies to the whole table including all of its records

## table.field

\* Name ▾  
Incident [incident] ▾  
Field ▾  
Caller

This rule applies to only one field on a record and in this case, the Caller field on an incident record

## table.\*

\* Name ▾  
Incident [incident] ▾  
\* ▾

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.

# Using the Wildcard Example

Wildcard ACL rules reduce the amount of rules required to control access.

Access	itil_admin	itil
change_request.None	read	read
change_request.*	read	
change_request.type		read

itil\_admin role sees:

Number	Short description	Type	State	Planned start date
CHG0000096	Change default router on unix201	Authorize	Authorized	2018-05-05 07:30:00
CHG0000095	Upgrade OWA-SD-01 to MS Windows Server 2016	Authorize	Authorized	2018-05-05 07:30:00
CHG0000094	Increase db_block_buffers from 5000 to 7500	Authorize	Authorized	2018-05-05 05:30:00

itil role sees:

Number	Short description	Type	State	Planned start date
X	Change default router on unix201	Normal	Normal	2018-05-05 07:30:00
X	Upgrade OWA-SD-01 to MS Windows Server 2016	Normal	Normal	2018-05-05 07:30:00
X	Increase db_block_buffers from 5000 to 7500	Normal	Normal	2018-05-05 05:30:00

One of the real benefits of using the wildcard 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. The role definition for the change\_request.\* rule grants access to all fields except those with a specific rule (change\_request.type in this example)

Without a wildcard rule, six separate rules would be required to control access. That is a significant more amount of rules to manage, should changes be needed in the future.



**Time**  
**15-20m**

- Update a role
- Provide application menu and module access for a specified role
- Create an Access Control rule to grant data permissions

## Lab 3.2: Application and Access Controls

As the **Data Security Manager**,

I want to **restrict access** to the Infinity Inventory application menu and modules and **control which fields can be updated**

So I can **ensure the integrity of the inventory data**.

# Application and Access Controls

**Lab  
3.2**

⌚15-20m

## Lab Objectives

You will achieve the following objectives:

- Update a role
- Provide application menu and module access for a specified role
- Create an Access Control rule to grant data permissions

## Scenario

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 well as controlling which fields the users can update.

### A. Modify the Infinity Table Role

*As the system administrator, you will act upon the requirements provided by Cloud Dimensions management to successfully secure Infinity device data.*

*The first step will be to modify the default role created during the creation of the Infinity table to create a new role for the group supporting the infinity product.*

1. Verify you are logged in as the **System Administrator**
2. Navigate to **System Security > Users and Groups > Roles**
3. 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.

4. Change the **name** to *u\_infinity\_support*
5. Type into the **Description** field: *Support role for the Infinity*

* Name	<input type="text" value="u_infinity_support"/>
Requires Subscription	<input type="text" value="Unspecified"/>
Description	<input type="text" value="Support role for the Infinity"/>

6. From the **Form Context Menu**, select **Insert**.

**Note:** Two infinity roles now exist which will allow for access provisioning in the future.

Name
Search
<i>u_infinity_support</i>
<i>u_infinity_user</i>

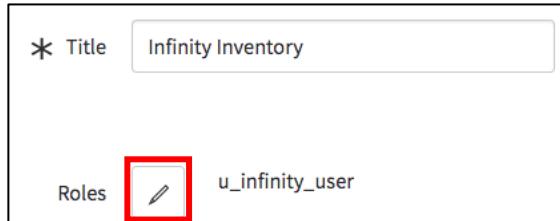
## B. Confirm and Modify Access to the Infinity Inventory Application

Now that the roles have been created, they can be used to configure access to the Infinity Inventory application.

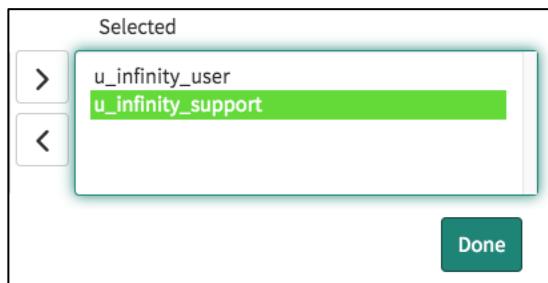
1. **System Definition > Application Menus.**
  2. Use any method to find and open the **Infinity Inventory** application menu.
- Discussion:** What are two methods for accessing the application menu form?
3. Verify the value next to the Roles field is **u\_infinity\_user**

* Title	<input type="text" value="Infinity Inventory"/>
Roles	<input type="text" value="u_infinity_user"/>

4. Click the **Edit User Roles** icon (pencil) next to the Roles field:



5. Add the **u\_infinity\_support** role from Available to Selected.



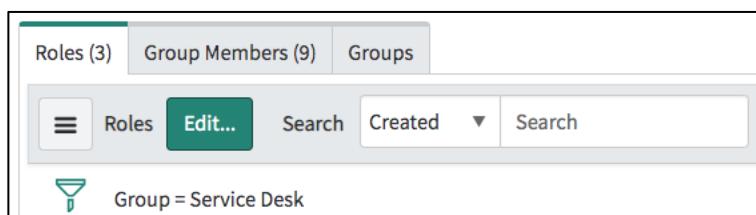
6. Click the **Done** button.

7. **Update** the Application Menu record.

## C. Update the Service Desk Group

*Roles have been created, but they are not yet assigned to a group. Let's fix that!*

1. Navigate to **User Administration > Groups**.
2. Locate and open the **Service Desk** group record.
3. Select the **Roles** tab, if needed, then click **Edit...**



4. Search for and select the **u\_infinity\_support** role.
5. **Save**.

6. The Service Desk Roles list should look like this:

Roles (4) Group Members (9) Groups			
Roles	Edit...	Search	Created ▾ Search
	Group = Service Desk		
	Created		Role
<input type="checkbox"/>		2017-09-15 02:25:58	itil
<input type="checkbox"/>		2017-09-15 02:25:58	filter_group
<input type="checkbox"/>		2019-07-26 19:58:32	u_infinity_support
<input type="checkbox"/>		2017-09-15 02:25:58	template_editor_group

**Note:** After adding the `u_infinity_support` role to the Service Desk group, multiple information messages will display indicating the role was added to the individual members of the group

## D. Test the Visibility Settings

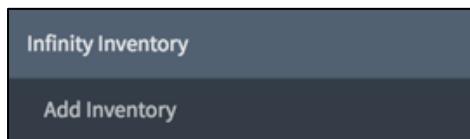
Now that the `u_infinity_support` role has been added to the Service Desk group, all members of the Service Desk group should have access to the Infinity Inventory application menu. Users who are not members of the Service Desk group should not have access.

We will test visibility of the Infinity Inventory application by impersonating users who are, and are not, assigned to the Service Desk group.

1. Impersonate **Rita Center**.
2. Enter **Infinity Inventory** in the **Filter navigator**.
3. Confirm **Rita Center** cannot access the **Infinity Inventory** application and its modules.

**Discussion:** Why is Rita unable to see the Infinity Inventory application?

4. Impersonate **Kevin Edd**.
5. Locate the **Infinity Inventory** application menu and expand it to confirm only **one** module displays:



**Discussion:** A module is missing from the application menu. Which one? Why?

6. Click on the **Add Inventory** module.

**Note:** Although **Kevin Edd** has inherited the **u\_infinity\_support** role because he is a member of the **Service Desk** group, which has access to the **Infinity Inventory** Application, the role does not currently have the rights for creating new data.

As a result, the **Infinity New** record page is blank when Kevin attempts to access it.

## E. Update Roles and Groups

---

**Access Control rules** can be created to allow users (who belong to a group with a certain role) access to work with a table's data, but first let us update Rita and Kevin's group permissions so Service Desk AND Infinity Customer Support group members are granted access to the application and all of its modules.

---

1. As System Administrator, navigate to **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. Select **Save**

**Note:** You have added the **u\_infinity\_user** role and its permissions under the **u\_infinity\_support** role. This means all users with the **u\_infinity\_support** role now automatically inherit the permissions found with **u\_infinity\_user**

---

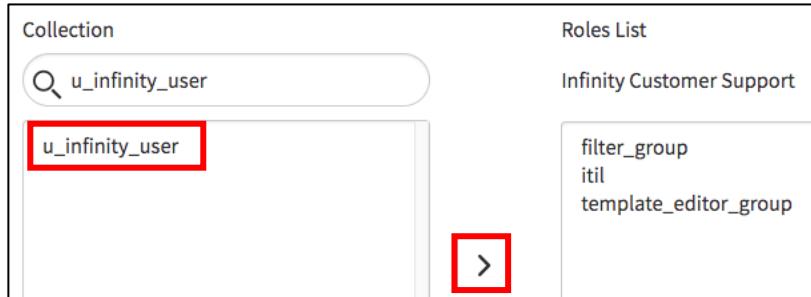
Now, the **u\_infinity\_support** role contains **u\_infinity\_user** role (which was created at the creation of the **Infinity** table and has read, write, update, and delete access to the **Infinity** application and its modules) so Kevin Edd can create new records.

Next, we need to add the **u\_infinity\_user** role to the **Infinity Customer Support** group so Rita (and other members of the group) will have access.

---

6. Navigate to **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.

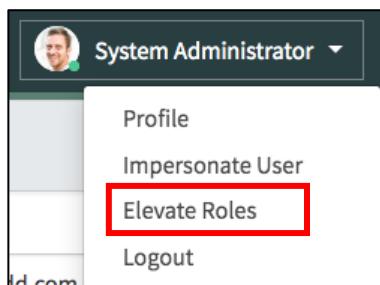
12. Confirm the **Infinity Inventory** application menu is accessible and its two modules display.

**Note:** Both Rita Center and Kevin Edd can now access both of the Infinity Inventory modules. Additionally, they can now create new records and update existing ones.

## F. Create an Access Control Rule

Now that the group permissions have been set and only the right groups of users can access the application, create an Access Control rule that allows only the System Administrator, and no other role, to update the Asset tag field on the Infinity record.

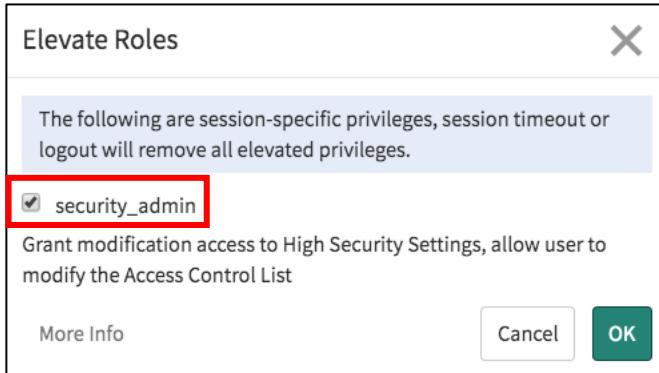
1. Impersonate the **System Administrator**.
2. Open the **User** menu.
3. Next, choose **Elevate Roles**:



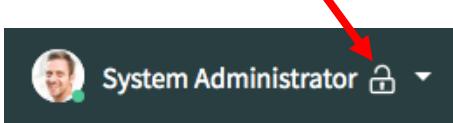
**Note:** The base system admin can elevate to a privileged role to have access to the features of High Security Settings.

If you grant additional users the admin role, they cannot elevate to a privileged role.

4. Select the checkbox next to **security\_admin**:



5. Select **OK**.
6. Notice the page refreshes and there is now an unlocked pad icon next to the user name on the User menu:

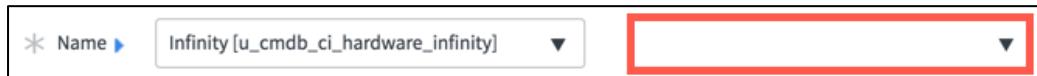


*With the elevated **security\_admin** role, you can now create access controls*

7. Navigate to **System Security > Access Control (ACL)**
  8. Filter the list of rules to find the four **u\_cmdb\_ci\_hardware\_infinity** rules.
- Discussion:** What are two ways you could locate the **u\_cmdb\_ci\_hardware\_infinity** rules you created most recently?
9. Open the rule with the **write** operation by selecting the information icon or the table name link:

Access Controls				
All > Name starts with u_				
Operation	Name	Type		
	<a href="#">u_cmdb_ci_hardware_infinity</a>	write	record	

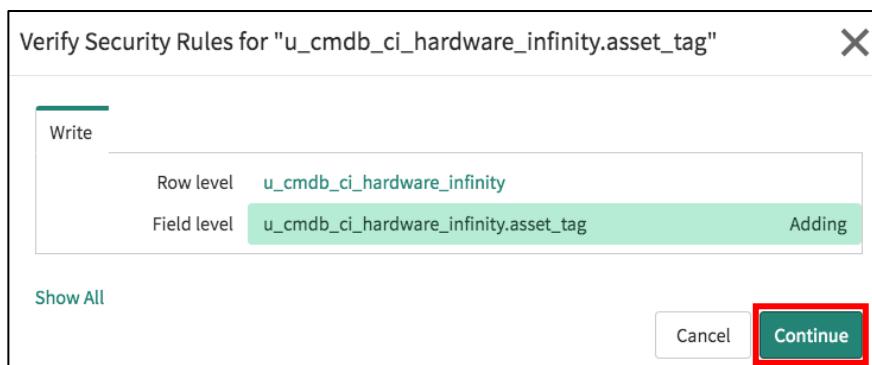
10. From the **Name** field, select the drop-down to the right of the selected value, **Infinity [u\_cmdb\_ci\_hardware\_infinity]**.



11. From the drop-down, select **Asset tag**.

12. Open the **Form Context Menu** and select **Insert and Stay**.

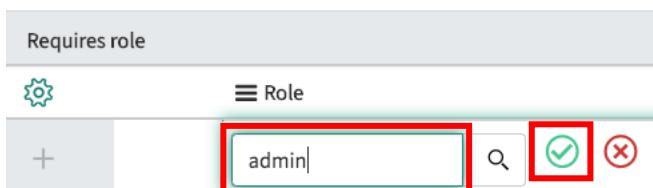
13. From the **Verify Security Rules** window, select **Continue**:



14. Scroll down to the **Definition** section and locate **Requires role**

15. Double click on **Insert a new row...**

16. Type **admin** and click the save icon to add the role.



17. **Update** the Access Control record

**Note:** The **u\_cmdb\_ci\_hardware\_infinity.asset\_tag** rule was created and added to the Access Control List.

## LAB VERIFICATION

1. Navigate to **System Security > Access Control (ACL)** and verify there are five access controls for the *u\_cmdb\_ci\_hardware\_infinity* table:

	Name	Operation	Type
<input type="checkbox"/>	<a href="#">u_cmdb_ci_hardware_infinity</a>	<a href="#">create</a>	<a href="#">record</a>
<input type="checkbox"/>	<a href="#">u_cmdb_ci_hardware_infinity</a>	<a href="#">delete</a>	<a href="#">record</a>
<input type="checkbox"/>	<a href="#">u_cmdb_ci_hardware_infinity</a>	<a href="#">read</a>	<a href="#">record</a>
<input type="checkbox"/>	<a href="#">u_cmdb_ci_hardware_infinity</a>	<a href="#">write</a>	<a href="#">record</a>
<input type="checkbox"/>	<a href="#">u_cmdb_ci_hardware_infinity.asset_tag</a>	<a href="#">write</a>	<a href="#">record</a>

2. Impersonate Rita Center
3. Navigate to **Infinity Inventory > Add Inventory**
4. Verify you can Add a new Infinity device.

Device Number	<input type="text"/>
Name	<input type="text"/>
Asset tag	<input type="text"/>
Device Version	<input type="text"/>

**Note:** All fields but **Asset tag** and **Email** should be editable. These same fields would also not be editable for existing records, for all users in Service Desk or Infinity Customer Support groups. As system administrator, all fields on the form are editable.

**With this lab, application and module access was adjusted, and an Access Control rule was created to limit permissions to table data.**

**This is not an easy topic nor lab – congratulations on completing it!**

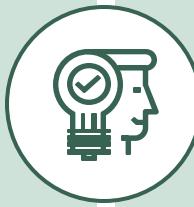
## Knowledge Check

What is specified in an access control list rule?

- **Object and Operation** being secured
- **Permissions** required to access the object

What types of permissions can be configured in an access control rule?

- Roles
- Conditions
- Script that sets the *answer* variable to true or false



In what order are access controls evaluated?

- **Table-level: most specific to most general** then **field-level: most specific to most general**

Which object grants access to all table records?

- **<table>.None**

Which elevated role is required to modify access control rules?

- **security\_admin**

## Section 3.3: Importing Data

### User Story

---

As the **Infinity Product Owner**,

I want a simple way to **import new inventory data** into ServiceNow and **update existing inventory data**

So I don't have to **manually maintain inventory items**.



# Importing Data: What are Import Sets?

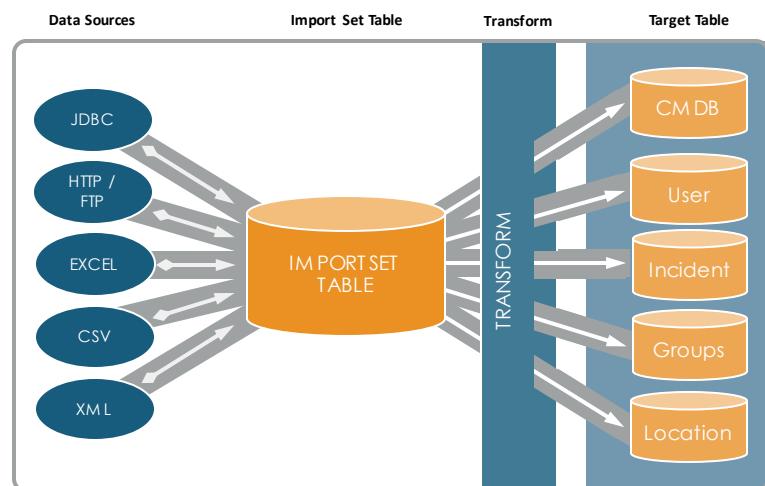
now

## Import Sets

An **Import Set** is a tool used to import data from various data sources, and map that data into ServiceNow tables



- HR Implementers may use Import Sets to copy a large number of HR Stories into the Development instance during implementation
- Security does not approve an integration with a data source vital to IT operations and use Import Sets to periodically update a significant number of records



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 (i.e. XML, CSV, Excel) 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.

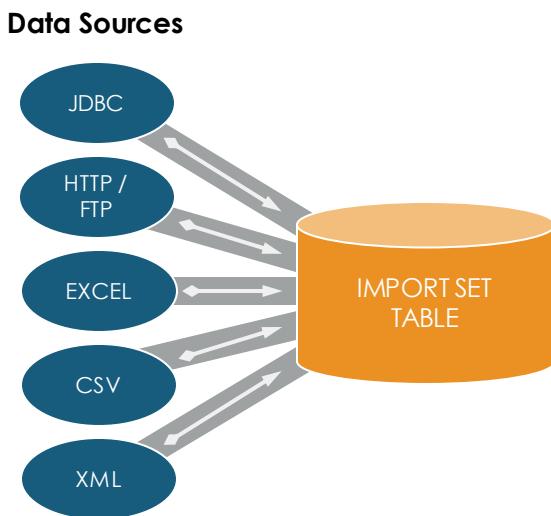
The **Import Set Table** acts as a staging area for records imported from a data source.

**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.

# Import Set Table



## Import Set Table

The **import set table** is a staging area for records imported from a data source

Fields on these tables are generated automatically based on imported data

## Importing Best Practices

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
- Easier to fix Inaccurate data before import

Before importing any data, it is important to understand what data you are bringing in and where that data should go.

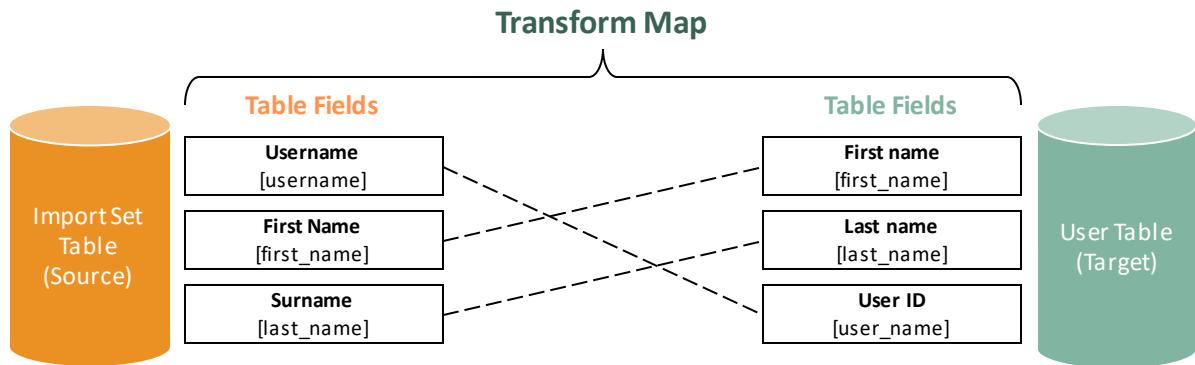
You should verify the data before you import it since bad data will complicate things later in the import and transform processes. Extra time spent planning and examining data before import will save time and potential problems later.

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

A **transform map** is a set of field maps that define the relationships between fields in an import set table and fields on a target table



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 ServiceNow 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 links 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

Coalescing a field (or multiple fields) means the field will be used as a unique key during imports



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 using the coalesce field(s), 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.

# DATA INTEGRITY

A **Data Policy** is a rule that enforces data consistency by setting fields as mandatory and/or read-only.

Data policies which have been configured for the table are applied when importing data.

Data Policies are applied to all data entered into the platform; form (UI), Import Sets, or Web Services.

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.

**NOTE:** Data Policies are not about security, they are about managing the integrity of the information stored in the database.



**Time**  
**20-25m**

- Modify a list layout as preparation
- Create an Import Set Table and Transform Map
- Transform multiple imports
- Clean up import data

## Lab 3.3: Import Sets

As the **Infinity Product Owner**,

I want a simple way to **import new inventory data** into ServiceNow and **update existing inventory data**

So I don't have to **manually maintain inventory items**.

# Import Data from Excel

**Lab  
3.3**

⌚20-25m

## Lab Objectives

You will achieve the following objectives:

- Modify a list layout as preparation
- Create an Import Set Table and Transform Map
- Transform multiple imports
- Clean up import data

**Required Resources:** *infinity-data.xlsx* and *infinity-updates.xlsx*

**Lab Dependency:** Requires the completion of Lab 3.1

## Scenario

In this lab, you will use Import Sets to load data that has been collected outside of ServiceNow into the Infinity table.

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 staging table and Transform Map
- Identify a coalesce (key) field to ensure that existing records are updated from the new imported data and not duplicated

In the data cleanup, you will clean up the Import Set Table's data.

## Part 1 - Initial Data Load

### A. Gather Excel (.xlsx) data for lab

1. Verify you have downloaded the two Infinity Excel files: **infinity-data.xlsx** and **infinity-updates.xlsx**

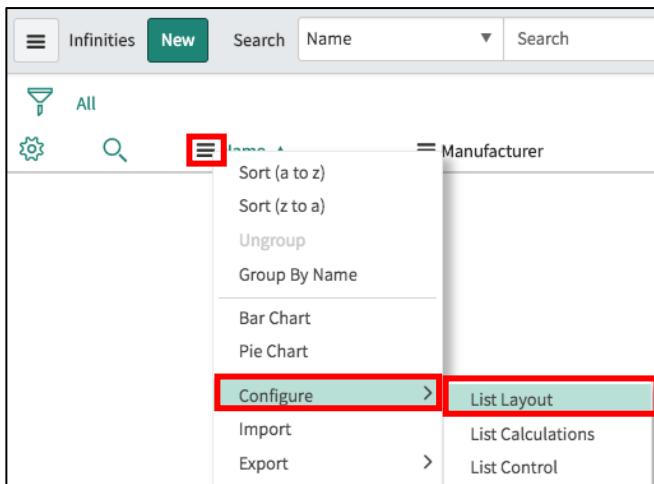
**Note:** In the event the files have not been downloaded, navigate to the ServiceNow Fundamentals Class knowledge base and select the Class Lab Files article

2. Open both the **infinity-data.xlsx** and **infinity-updates.xlsx** files to review the contents of each file including columns, fields, and data types.

### B. Modify the Infinity All Devices List Layout

Modify the Infinity All Devices list layout to ensure we can clearly examine the new records after they are imported into ServiceNow.

1. As the System Administrator, navigate to **Infinity Inventory > All Devices**
2. Open the **Column Context 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:

Selected
Device Number
Device Version
Name
Owned by
Location
Status
Support group
Updated

4. Click **Save**.
5. Verify the order of the fields in **Infinities** list header:

The screenshot shows the header of a list view for 'Infinities'. At the top left are buttons for 'New' and 'Search'. To the right is a search bar with a dropdown arrow and a placeholder 'Search'. Below the search bar are two filter icons: a funnel for 'All' and a gear for 'Filter'. To the right of these are eight column headers, each preceded by a sorting icon: 'Device Number', 'Device Version', 'Name ▲', 'Owned by', 'Location', 'Status', 'Support group', and 'Updated'.

## C. Create a New Import Set

1. System Import Sets > Load Data

**Note:** The following steps (process) can be completed by any user with the role `import_admin` or `import_set_loader` and `import_transformer`.

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**.

- Review the Progress screen, you should see **7 inserts**:

Progress	
Name	ImportProcessor
State	Complete
Completion code	Success
Message	Processed: 7, inserts 7, updates 0, errors 0, empty and ignored 0, ignored errors 0 (0:00:00.940)

## D. Validate Data in Import Set

- To verify the data in the new Import Set, in the **Next Steps...** section of the **Progress** screen, select the **Loaded data** link:

Next steps...

[Import sets](#) Go to the import sets for this data load

[Loaded data](#) Go to the newly imported data inside the staging table: u\_infinity\_imports

[Create transform map](#) Create a transform map for the newly staged data

- Confirm the seven **Infinity Imports** records are shown with a state of Pending:

Infinity Imports				
	New	Search	Set	Search
All				
<input type="checkbox"/>	<a href="#">(i)</a>	3 ISET0010001	Pending	
<input type="checkbox"/>	<a href="#">(i)</a>	1 ISET0010001	Pending	
<input type="checkbox"/>	<a href="#">(i)</a>	5 ISET0010001	Pending	
<input type="checkbox"/>	<a href="#">(i)</a>	2 ISET0010001	Pending	
<input type="checkbox"/>	<a href="#">(i)</a>	6 ISET0010001	Pending	
<input type="checkbox"/>	<a href="#">(i)</a>	4 ISET0010001	Pending	
<input type="checkbox"/>	<a href="#">(i)</a>	0 ISET0010001	Pending	

**Note:** You may notice the records in a different order than shown

## E. 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\_hardware\_infinity]**

3. Save (not Submit) the record.

4. Scroll to Related Links, then click **Auto Map Matching Fields**.

**Note:** A verification message displays at the top of your form.

5. Verify four fields are auto mapped: **Name, Device Number, Device Version, and Support Group**:

Field Maps (4)			
Transform Scripts			
Field Maps		New	
<input type="checkbox"/>	Source field	Target field	Coalesce
<input type="checkbox"/>	u_support_group	support_group	false
<input type="checkbox"/>	u_device_number	u_device_number	false
<input type="checkbox"/>	u_device_version	u_device_version	false
<input type="checkbox"/>	u_name	name	false

**Note:** Your field order may be different.

6. In the Related Links list, select **Mapping Assist**

7. From the Source: Infinity Imports staging table, move **Device Owner** and **Owner Location** into the Field Map.

The screenshot shows the 'Mapping Assist' interface. On the left, under 'Source: Infinity Imports', the 'Owner Location' field is selected and highlighted with a red box. An 'Add' button with a right-pointing arrow is also highlighted with a red box. On the right, under 'Field Map', there is a table with two columns: 'Support Group' and 'Support group'. The 'Owner Location' field has been moved to this table and is highlighted with a red box. The 'Add' button is located between the source list and the field map table.

8. From the Target: Infinity table, move **Owned by** and **Location** into the Field Map
9. Verify the Source: and Target: fields are mapped properly

Field Map	
Device Version	Device Version
Support Group	Support group
Name	Name
Device Number	Device Number
Device Owner	Owned by
Owner Location	Location

10. Select **Save** and verify there are now six mapped fields:  
**Name, Device Number, Device Version, Support Group, Owned by, and Location**

Field Maps			
	Source field	Target field	Coalesce
<input type="checkbox"/>	<i>u_support_group</i>	support_group	false
<input type="checkbox"/>	<i>u_device_number</i>	u_device_number	false
<input type="checkbox"/>	<i>u_device_version</i>	u_device_version	false
<input type="checkbox"/>	<i>u_name</i>	name	false
<input type="checkbox"/>	<i>u_device_owner</i>	owned_by	false
<input type="checkbox"/>	<i>u_owner_location</i>	location	false

## F. Run the Transform

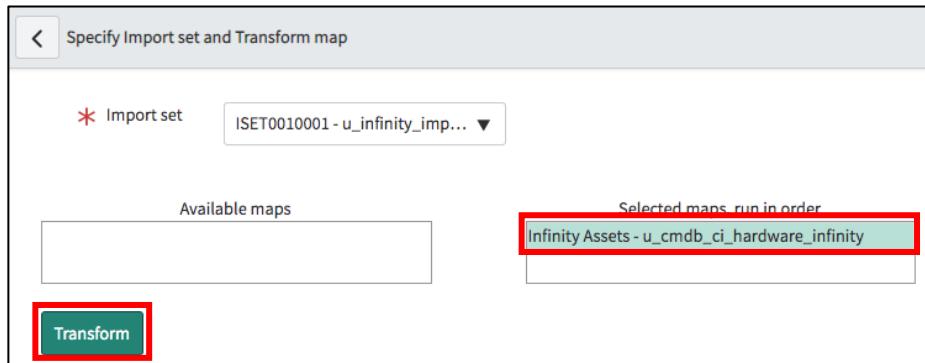
1. From Related Links of the Infinity Assets Transform Map, select **Transform**

Table Transform Map  
Infinity Assets

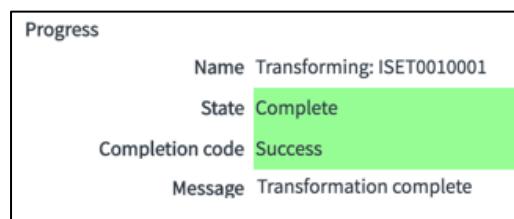
Related Links

- [Auto Map Matching Fields](#)
- [Mapping Assist](#)
- [Transform](#)
- [Index Coalesce Fields](#)

- Verify the **Infinity Assets – u\_cmdb\_ci\_hardware\_infinity** map is selected



- Select the **Transform** button
- The **Progress** screen displays the transformation confirmation messages:



## G. Verify Infinity Inventory import

- Infinity Inventory > All Devices.**
- The list should display seven total records.

	Device Number	Device Version	Name	Owned by	Location	Status	Support group
<input type="checkbox"/>	<a href="#">CDE0100106</a>	AP01	Infinity Alpha Prototype	<a href="#">Kevin Edd</a>	<a href="#">153 South Sierra Avenue, Solana Beach, CA</a>	Installed	<a href="#">Service Desk</a>
<input type="checkbox"/>	<a href="#">CDE0100105</a>	AP01	Infinity Alpha Prototype	<a href="#">Winnie Reich</a>	<a href="#">9249 Cicero Avenue, Oak Lawn, IL</a>	Installed	<a href="#">Service Desk</a>
<input type="checkbox"/>	<a href="#">CDE0100103</a>	TD01	Infinity Testing Device	<a href="#">Beth Anglin</a>	<a href="#">6304 Northwest Barry Road, Kansas City, MO</a>	Installed	<a href="#">Service Desk</a>
<input type="checkbox"/>	<a href="#">PAR0100099</a>	TD01	Infinity Testing Device	<a href="#">Jon Floyd</a>	<a href="#">10335 Lee Hwy, Fairfax, VA</a>	Installed	<a href="#">Infinity Customer Support</a>
<input type="checkbox"/>	<a href="#">CDE0100101</a>	TD01	Infinity Testing Device	<a href="#">Kevin Edd</a>	<a href="#">153 South Sierra Avenue, Solana Beach, CA</a>	Installed	<a href="#">Service Desk</a>
<input type="checkbox"/>	<a href="#">CDE0100099</a>	TD01	Infinity Testing Device	<a href="#">Winnie Reich</a>	<a href="#">9249 Cicero Avenue, Oak Lawn, IL</a>	Installed	<a href="#">Service Desk</a>
<input type="checkbox"/>	<a href="#">CDE0100102</a>	TD02	Infinity Testing Device	<a href="#">Kevin Edd</a>	<a href="#">153 South Sierra Avenue, Solana Beach, CA</a>	Installed	<a href="#">Service Desk</a>

## Part 2 - Incremental Data Load

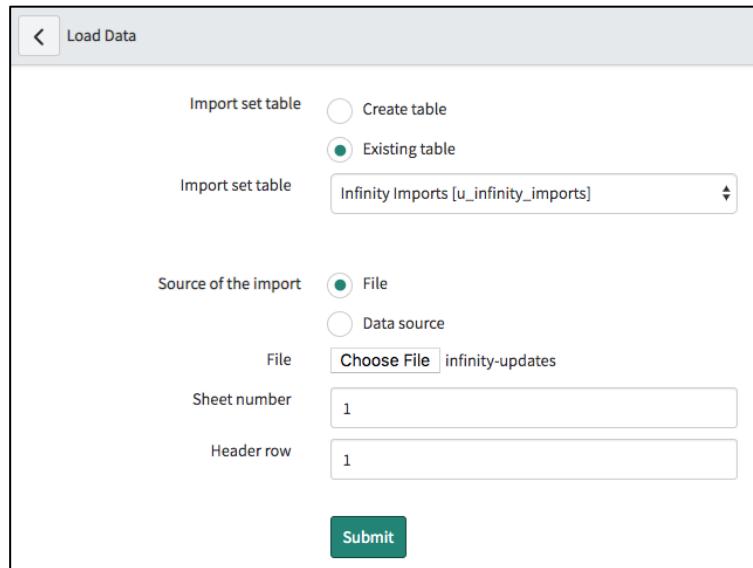
Using the existing import set table (Infinity Imports) and existing transform map, you will load additional data from the **infinity-updates.xlsx** file.

### A. 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** and select  
**infinity-updates.xlsx**

4. Click **Submit**.
5. You should see **19 inserts**.



The screenshot shows the 'Load Data' dialog box. In the 'Import set table' section, 'Existing table' is selected for 'Create table' and 'Infinity Imports [u\_infinity\_imports]' is selected for 'Import set table'. In the 'Source of the import' section, 'File' is selected. Below it, 'File' is set to 'infinity-updates', 'Sheet number' is '1', and 'Header row' is '1'. A 'Submit' button is at the bottom.

**Note:** 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, they key field should be defined to ensure existing records are updated rather than added as new records.

To add a coalesce field, we will need to modify the existing Transform Map.

### B. Define the Coalesce Field

1. **System Import Sets > Administration > Transform Maps**
2. Open the **Infinity Assets** Transform Map.
3. Scroll to the **Field Maps** Related Lists.

**Note:** The device number (**u\_device\_number**) field seems to be the best candidate as the coalesce because it will contain a unique value. If a match is found for the value of this field, then the record will be updated instead of a new record being created.

- From the **u\_device\_number** row, in the **Coalesce** column, double-click the word **false**, then select **true** from the list:



- Select **Save** (green checkmark) to update the value to true.

**Note:** A system message displays at the top of the form.

## C. Prepare and Run the Transform

- Under the Related Links section, select the **Transform** link.
- In the Selected maps box, verify you are using the correct Transform Map: **Infinity Assets – u\_cmdb\_ci\_hardware\_infinity**
- Select **Transform**
- Confirm the transformation complete message is displayed.

## D. Verify Infinity Inventory Import Updates

- Infinity Inventory > All Devices.**
- Confirm there are **19 total records** in the Infinities list.



**Note:** Notice that some of the records have an updated 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

- Verify the six (6) Field Maps records are visible in the *Infinity Assets Transform Map* and the Coalesce field for the u\_device\_number is true.

Source field	Target field	Coalesce
u_support_group	support_group	false
u_device_number	u_device_number	true
u_device_version	u_device_version	false
u_name	name	false
u_device_owner	owned_by	false
u_owner_location	location	false

- Review the imported Infinity Inventory data ( **Infinity Inventory > All Devices** ) and verify there are 19 records.

Device Number	Device Version	Name	Owned by	Location	Status	Support group	Updated
CDE0100107	AP01		Megan Burke	3121 High Point Road, Greensboro, NC	Installed	(empty)	2019-17:05:05
CDE0100106	AP01	Infinity Alpha Prototype	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	Service Desk	2019-14:14:49
CDE0100105	AP01	Infinity Alpha Prototype	Winnie Reich	9249 Cicero Avenue, Oak Lawn, IL	Installed	Service Desk	2019-14:14:49
CDE0100103	TD01	Infinity Testing Device	Beth Anglin	6304 Northwest Barry Road, Kansas City, MO	Installed	Service Desk	2019-14:14:49
PAR0100103	TD01	Infinity Testing Device	Jon Floyd	10335 Lee Hwy, Fairfax, VA	Installed	Infinity Customer Support	2019-17:05:05
CDE0100109	TD01	Infinity Testing Device	Helene Iberg	Bockenheimer Landstraße 223, Frankfurt	Installed	Service Desk	2019-17:05:05
CDE0100111	TD01	Infinity Testing Device	Trey Tout	30 Katharinenstr, Hamburg	Installed	Service Desk	2019-17:05:05
PAR0100101	TD01	Infinity Testing	Jon Cloud	10335 Lee Hwy, Fairfax, VA	Installed	Infinity Customer	2019-

**Congratulations on completing the Import Sets lab!**

**Because of you, Cloud Dimensions has visibility of the current inventory previously stored in the legacy system – way to go!**

## Knowledge Check

Which module is used as the first step for importing data?

- Load Data

What are the steps for importing data using an import set?

- Load the data
- Create a transform map
- Transform the data
- Clean up the import table



Which option allows you to update existing target table records when importing data from an import set?

- Coalesce

Which tool is used to determine relationships between fields in an import set and an existing table?

- Transform Map

## Section 3.4: CMDB

### User Story

---

As the **Configuration Manager**,

I want a way to **track the Infinity devices and their relationships** with other configuration items

So I can **understand the impact** of adding the configuration items to our existing infrastructure.



# The CMDB and Configuration Items

## CMDB

The **Configuration Management Database** 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.

Access to the CMDB tables and underlying data requires certain permissions, such as the following roles:

- asset
- itil
- itil\_admin



## Configuration Items

**Configuration Items** can be tangible or intangible devices or applications in the CMDB such as firewalls, computers, email services, and business services

### Computers



### Devices on the network



### Applications



### Business services



The Configuration Application provides core functionality for the Configuration Management Database (CMDB), including modules for hardware and other 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.

A configuration item 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.

ServiceNow's CMDB, in contrast to a static list, not only tracks the CIs within your platform, but also the relationships between those items.

Two key CMDB tables are **Configuration Item [cmdb\_ci]** which contains CI data, and **CI Relationship [cmdb\_rel\_ci]** which contains CI relationship data.

# Configuration Item Form

**CI Attributes**

**Relationships to other Cls**

**Toggle between CI Health Dashboard view and Form view**

**Related Items**

**Related Items toolbar**

Click the reference icon (i) to the right of the **Configuration item** field to be redirected to the selected configuration item 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.

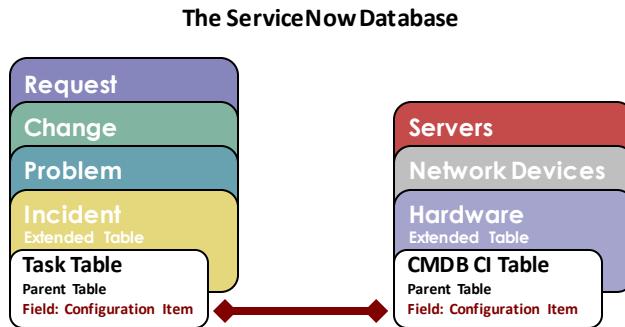
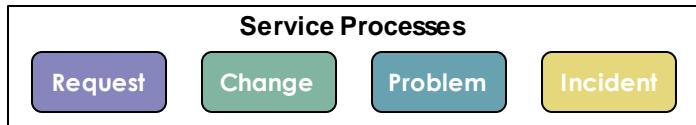
Toggle between the CI Health Dashboard view and the form view using the options in the form title bar.

The form also contains information about the relationships between items. Search for configuration items, 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 > Relationships > Suggested Relationships**.

You can view relationships between the current item and other items. An advanced feature is the Related List in configuration item records which displays additional components.

# Service Processes: Using the CMDB



## Benefits

Typically in businesses, a high percentage of incidents are caused by failed changes

Benefits of having an accurate and up-to-date CMDB include:

- Locating failed changes and associated incidents
- Facilitating quick analysis of impact, helping reduce or eliminate downtime

The Configuration Management Database 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.

# CI Relationship Editor

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

Name	Manufacturer	Location	Description	Class	Updated	Maintenance schedule
*ANNIE-IBM	Lenovo	815 E Street, San Diego, CA		Computer	2017-08-16 01:46:29	
*ASSET-IBM	Lenovo	3 Whitehall Court, London		Computer	2017-08-16 01:46:32	

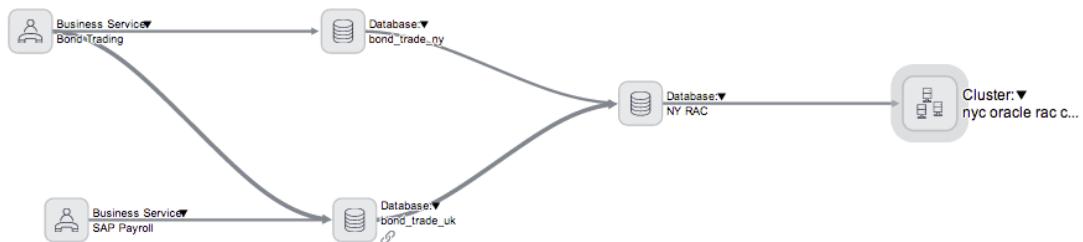
Use the CI relationship editor to create configuration item relationships. It is accessed from the Related Items toolbar on a configuration item form.

When you use the relationship editor, the CI record which the editor was launched is designated as the base configuration item. You can then select one or more items to include in the relationship. Depending on the selected relationship type, the base CI can become the parent CI or the child CI in a new relationship.

## Configuration Items: Dependency View

**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

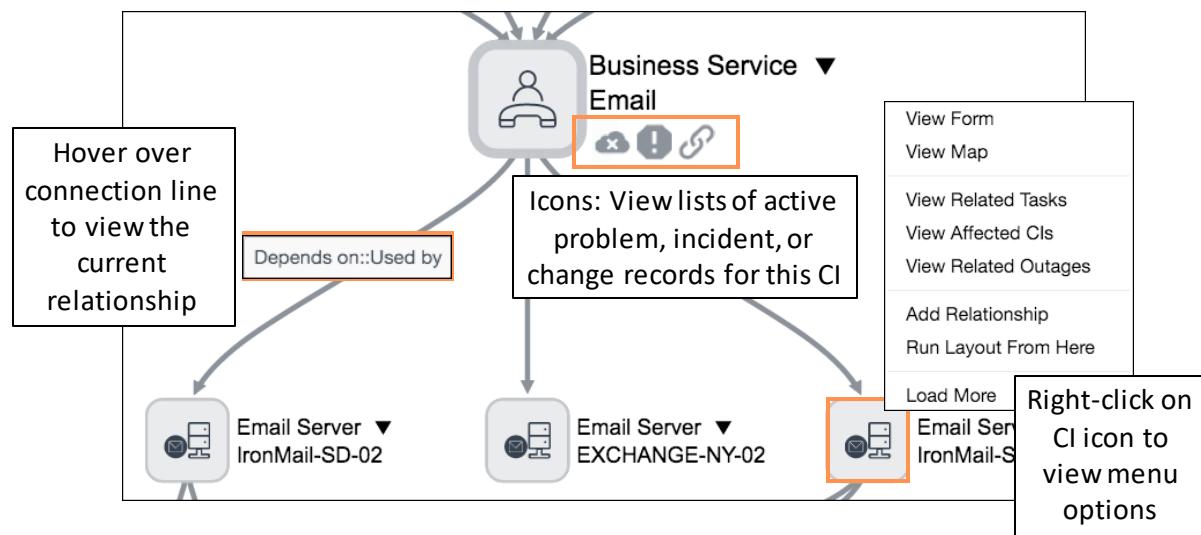


The Dependency View mapping interface is powered by D3 and Angular technology, providing a modern interactive graphical interface to visualize configuration items and their relationships.

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 items that the email service is dependent on.

# Dependency View: Map Icons

now.



In a Dependency View, icons and glyphs indicate whether a CI has an active, pending issue. You can investigate the tasks that are connected to a configuration item to get more details. The map collapses and expands clusters to make them easier to view.

Understanding the dependencies and other relationships among configuration items will enhance the operational delivery of incident, change, and problem management processes.



**Time**  
**10-15m**

- Create a new CI class in the CMDB for Infinity devices
- Define CI relationships

## Lab 3.4: CMDB

As the **Configuration Manager**,

I want a way to **track the Infinity devices and their relationships** with other configuration items

So I can **understand the impact of adding the configuration items to our existing infrastructure.**

# Configuration Management Database (CMDB)

**Lab  
3.4**

⌚10-15m

## Lab Objectives

You will achieve the following objectives:

- Create a new CI class in the CMDB
- Define CI relationships

## Scenario

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.

## A. 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.

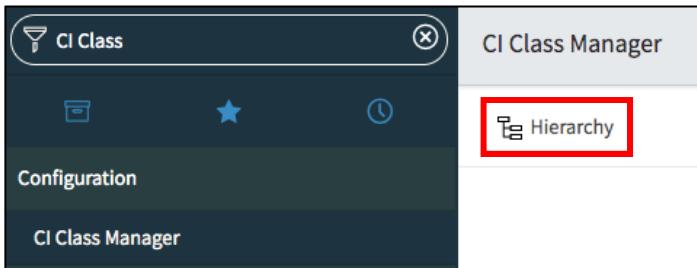
In Lab 1.3, a new CI Class was defined for the Infinity – categorizing it as a Hardware CI because of the `cmdb_ci_hardware` table extension.

### 1. Impersonate Darrel Tork.

**Note:** This user is the Configuration Manager of Cloud Dimensions. They were provided the `itil` role, in order to access the **CI Class Manager**. If your company does not have a Configuration Manager, this responsibility could fall onto the system administrator.

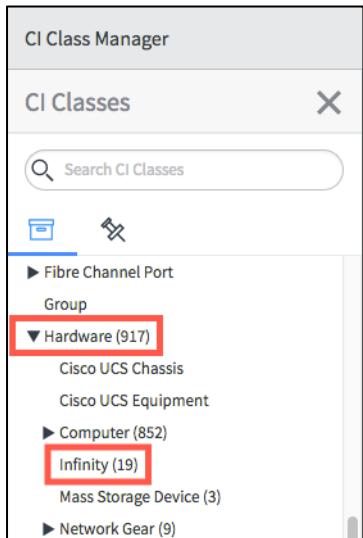
2. Navigate to Configuration > CI Class Manager.

3. Select the **Hierarchy** button:



4. Search for *Infinity* in **Search CI Classes**

**Note:** There are 19 records in the *Infinity* CI Class. These were created when the *u\_cmdb\_ci\_hardware\_infinity* table was populated with the imported data (Lab 3.3).



**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.

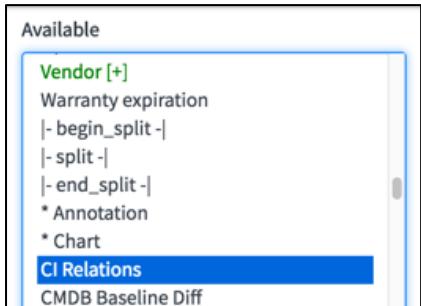
## B. Add a New Infinity Device to the CMDB

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

1. **Infinity Inventory > Add Inventory.**

2. From the **Form Context Menu** select **Configure > Form Layout**.

3. Scroll down the list of Available fields to locate **CI Relations**:



**Note:** 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. Select the **Save** button.
6. Fill out the Infinity New Record form as shown:

Device Number: **CDE0100999**  
Name: **Infinity Beta Prototype**  
Device Version: **BP01**  
Support Group: **Service Desk**  
Installed: **[today's current date and time]**  
Owned by: **Darrel Tork**  
Email: **darrel.tork@cloudd.com** (auto-fills)

7. **Submit.**

## C. Create a New Suggested CI Relationship

Switching back to system administrator, you will create a new suggested CI relationship between Infinity and the Infinity Media Server, using the Infinity record created by Darrel.

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. Impersonate **System Administrator**.
2. **Configuration > Relationships > Suggested Relationships**.

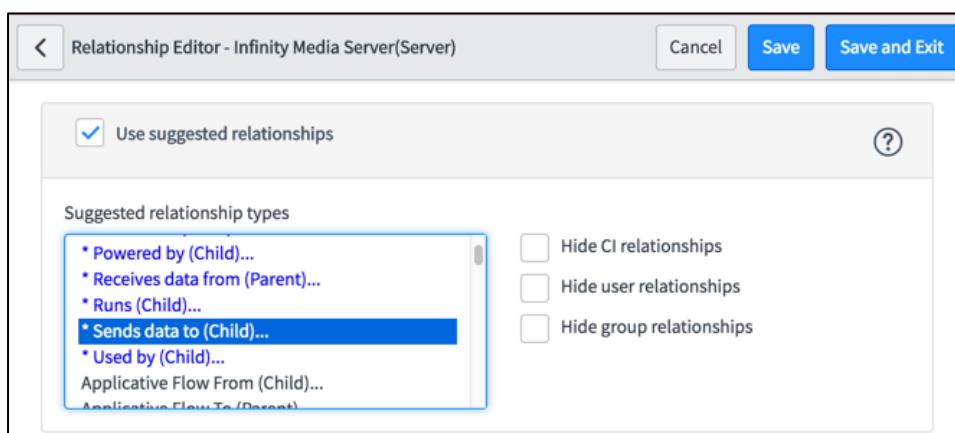
3. Select **New**.
4. Fill out the form as shown:

Base class: **Infinity**  
Relationship: **Receives data from (parent)**  
Dependent class: **Server**

5. **Submit**.
6. **Configuration > Servers > All**.
7. **New**.
8. Name: **Infinity Media Server**.
9. **Save**.
10. Scroll down to the Related Items section, and select the **Add CI relationship** icon:



11. In the Suggested relationship types field, select **\* Sends data to (Child)...**



12. From the **Configuration Items** section, use the Updated field to sort the records by last updated.

13. Use the checkbox to select the **Infinity Beta Prototype** record:

Configuration Items					
	Name	Manufacturer	Location	Description	Class
<input checked="" type="checkbox"/>	Infinity Beta Prototype				Infinity

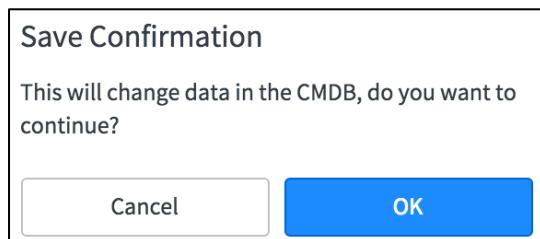
**Note:** While only the **Sends data to (Child)** relationship was created, suggested relationships also accounts for the converse relationship: **Receives data from (Parent)**.

14. Scroll down to the Relationships section and add **Infinity Media Server** by selecting the **Create new relationships** icon (+).

Relationships		
Type	Parent	Child
<input type="checkbox"/> Receives data from::Sends data to	Infinity Beta Prototype	Infinity Media Server

15. Select the **Save and Exit** button.

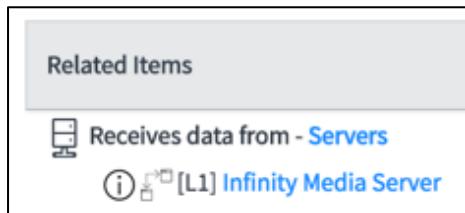
16. A Save Confirmation pop-up message may display. Select **OK**.



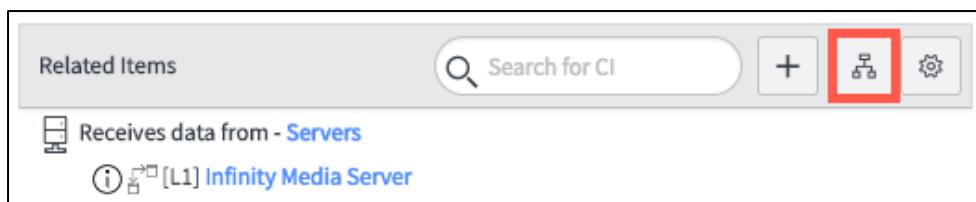
## LAB VERIFICATION

### New Infinity Media Server CI and Relationships

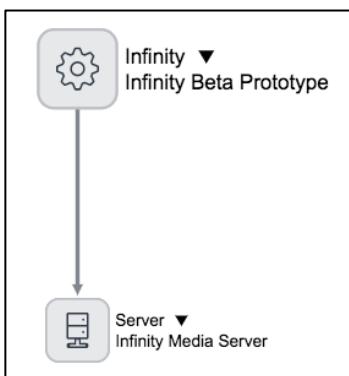
1. **Infinity Inventory > All Devices.**
2. Locate and open the **Infinity Beta Prototype** record: **CDE0100999**.
3. Verify the Related Items section looks like this:



4. Open the Dependency View by Selecting 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.

**Excellent work! You have learned how to create a new CI class, add a CI to the CMDB, and use Suggested Relationships to create relationships between configuration items!**

## Knowledge Check

What is defined as a series of tables containing assets and business services controlled by a company?

- Configuration Management Database (CMDB)

What is a CI?

- Configuration Item
- Can be tangible (hardware, software, servers, etc.) or intangible (business services, email, etc.)



Which ServiceNow product can be used to populate the CMDB?

- Discovery

What are the three key tables in the CMDB?

- cmdb
- cmdb\_ci
- cmdb\_rel\_ci



# Module 4: Self-Service and Process Automation

now.

User Interface and Navigation

Collaboration

Database Administration

**Self-Service and Automation**

Introduction to Development

Capstone Project

## Module Objectives

- Create a knowledge base article by importing a Word document
- Explain how to locate and approve knowledge base articles
- Define, apply, and test user criteria on a knowledge base
- Demonstrate how to create a catalog item with variables
- Use the Try It functionality to test catalog item ordering
- Create a Flow Designer Flow and associate it with a service catalog item
- Test the flow by ordering a catalog item

## Labs and Activities

- Lab 4.1: Knowledge Management
- Lab 4.2: Create a Catalog Item
- Lab 4.3: Create a Flow Designer Flow

## Section 4.1: Knowledge Management

### User Story

---

As the **Knowledge Manager**,

I want to be able to **publish knowledge articles** visible to only Cloud Dimensions employees,

So I can advertise Infinity employee testing **opportunities and guidelines** to Cloud Dimensions employees.



# What is Knowledge Management?

**Knowledge Management** allows users to create, categorize, review, approve, and browse important information in a centralized location that is shared by the entire organization.

Knowledge content exists within a Knowledge Base, which is managed by one or more Knowledge Managers.

Administrators and those with the **knowledge\_admin** role have the ability to manage multiple Knowledge Bases.



## Real World Use Cases

- *HR administrators can limit access to Knowledge articles with User Criteria, for example Benefits for EMEA employees are only visible to employees who reside in EMEA*

- *Include HR policies, calendars, and detailed instructions for reporting violations to reduce security cases*

- *In conjunction with Event Management, Knowledge Base articles containing resolution instructions can be generated from Events to fix an issue with a CI*

With Knowledge Management, each organization can have their own Knowledge Base (KB) with flexible controls over who can see the information and who can help develop its content.

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 or search for articles, sorting by relevancy, most recently updated, and the number of views.

Once an article has a category and is accessible in the knowledge base, there are a number of features that allow the organization's users to provide their feedback whether adding comments to the article or flagging it, which will bring the article to the attention of the KB administrators.

Some ServiceNow applications, such as Incident, allow contextual searching of Knowledge Base content. This gives users the ability to troubleshoot their issue before submitting an incident by displaying potential relevant articles.

# Knowledge Base Architecture

The Knowledge homepage displays knowledge articles and social questions (Q&A) organized by **Knowledge Base** and **Category**

The screenshot shows the ServiceNow Knowledge Base interface. On the left, under 'Knowledge Bases', there are two items: 'IT' (3 Questions and 31 Articles) and 'Social QA' (0 Questions and 0 Articles). Below this is a 'Featured Content' section with a message about Sales Force Automation being down. On the right, under 'Categories', the 'Applications' category is selected, showing a list of sub-categories: Microsoft, Devices, Email, IT, Operating Systems, and Suppliers. Below this list are two knowledge articles: 'Managing Settings in Internet Explorer 10 for Windows 8' and 'Excel Functionality'. A callout box labeled 'IT Knowledge Base' points to the 'IT' category in the sidebar.

From the homepage, users with the correct permissions can import a Word document to a Knowledge Base using the **Import Articles** button, create a new article using the **Create an Article** button, or ask a question using the **Post a Question** button.

Administrators can create multiple Knowledge Bases and assign them to individual managers responsible for controlling the behavior and organizational scheme of each Knowledge Base. Every Knowledge Base can have unique lifecycle workflows, user criteria, category structures, and management assignments.

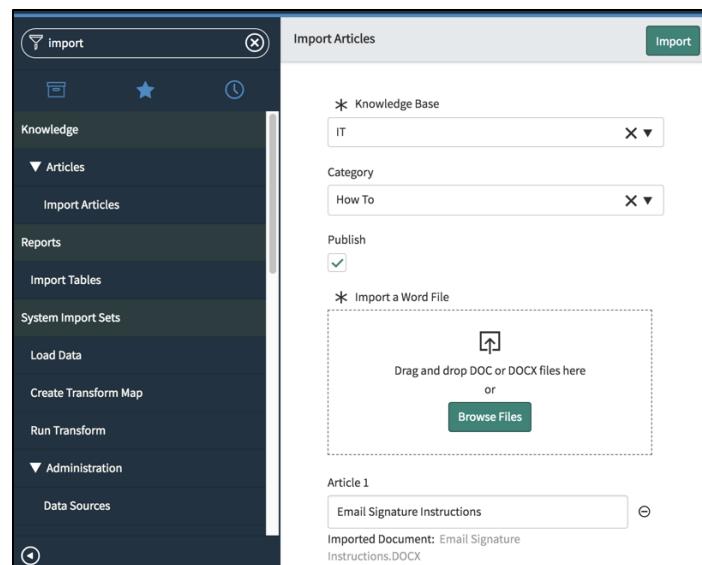
## Category Hierarchy:

- 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 Base: Import a Word document

To import a Word document into the Knowledge base:

1. Navigate to **Knowledge > Articles > Import Articles**
2. Select the **Knowledge Base**
3. Select the **Category\***
4. Check the **Publish** check box\*
5. Add the Word file (drag and drop or browse for the file)
6. Click **Import**



When importing a Word document, the following styles and elements are preserved:

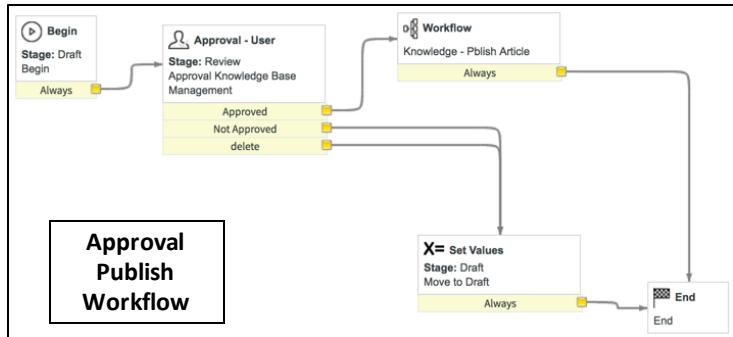
- Titles
- Headings
- Images (Images might not be aligned exactly as in the Word document you import)
- Links
- Bold text
- Italic text
- Underlined text
- Ordered and unordered lists
- Tables

\*Optional steps. Check the **Publish** check box to initiate the publishing workflow after the import completes.

**NOTE:** If the publish checkbox is not visible, the system administrator will need to modify a system property. To show the **Publish** check box, navigate to **Knowledge > Administration > Properties** and activate the **Show publish checkbox on the knowledge import pop-up** property.

## Knowledge Base: Workflows

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 Base workflows available in the ServiceNow baseline 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 – Approval Retire:** Requests approval from a manager of the Knowledge Base before moving the article to the retired state. The workflow is canceled and the article remains in the published 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 – Publish Knowledge:** A subflow that moves the knowledge article to the published state. You can use this subflow when defining your own workflow.
- **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.

# Knowledge Security: User Criteria

now.

**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 a 'User Criteria' record. The title bar says 'User Criteria' and 'The ACME North America HR Department'. The top right has standard buttons: Update, Delete, and navigation arrows. On the left, there's a sidebar with tabs: 'Name' (selected), 'Application', 'Active' (checked), 'Companies', 'Locations', 'Departments', and 'Advanced'. Under 'Name', it says 'The ACME North America HR Department'. Under 'Application', it says 'Global'. Under 'Active', there's a checked checkbox. Under 'Companies', it lists 'ACME North America' with a lock icon. Under 'Locations', there's a lock icon. Under 'Departments', it lists 'HR' with a lock icon. Under 'Advanced', there's a checked 'Match All' checkbox.

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 outcomes 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. User Criteria records are accessed from the **Can read** or **Can contribute** related lists.

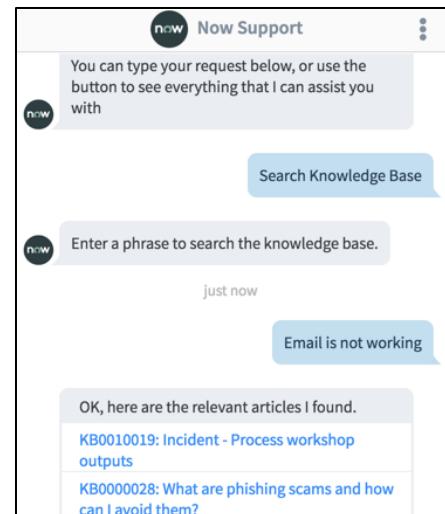
When creating user criteria, the **Match All** check box to determine whether all elements from each populated criteria field must match. If selected, only users who match all criteria are given access. If cleared, the user must meet one or more of the set criteria to be given access. By default, this check box is cleared so that any condition met provides a match.

## Virtual Agent

Virtual Agent is a conversational bot platform that provides assistance to help users obtain information, make decisions, and perform common work tasks

This includes information stored in the Knowledge Base, Service Catalog, and more

At any time, users have the option to switch to speak with a human agent for assistance, making sure they receive the help they need



*Learn more about Virtual Agent in Now Learning*

The biggest benefit to implementing a virtual agent is so your users can get immediate help, day or night.

Offer a personalized customer experience with a virtual agent by automating typical Tier 1 support tasks to be accomplished, including:

- Answering FAQs
- Providing tutorial (“how to”) information
- Querying or updating records – for example: get the status on cases or incidents
- Gathering data, such as attachments, for the agent
- Performing diagnostics
- Resolving multi-step problems

Virtual Agent offers a web-based interface available for Service Portal, iOS and Android mobile environments, and also supports third-party messaging applications through ServiceNow adapters for Slack and Microsoft Teams.

Learn more about Virtual Agent by visiting the ServiceNow Developer site, [developer.servicenow.com](https://developer.servicenow.com) and selecting **Learn > Training**



**Time**  
**10-15m**

- 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 4.1: Knowledge Management

As the **Knowledge Manager**,

I want to be able to **publish knowledge articles** visible to only Cloud Dimensions employees,

So I can advertise Infinity employee testing **opportunities and guidelines** to Cloud Dimensions employees.

# Knowledge Management

Lab  
4.1

10-15m

## Lab Objectives

You will achieve the following objectives:

- 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

## Scenario

After a few positive rounds of Infinity testing, the product has been greatly improved and the testing audience is ready to be expanded to include interested Cloud Dimensions employees.

Working with members of the Human Resources group, the Infinity testing coordinator will provide guidelines for enrolling in Infinity testing to be published in the HR knowledge base.

Appropriate members of the HR group will be granted the appropriate authoring permissions by the system administrator so that they may create, review, and publish articles.

The system administrator will assist with ensuring the article remains secure and accessible only by Cloud Dimensions employees.

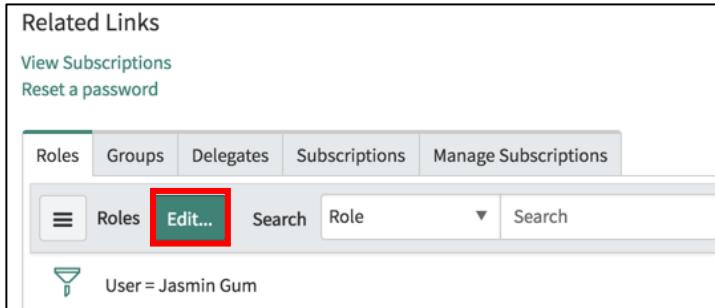
**Required Resource(s):** *Infinity\_Open\_Enrollment.docx*

## A. Assign Role to Knowledge Manager

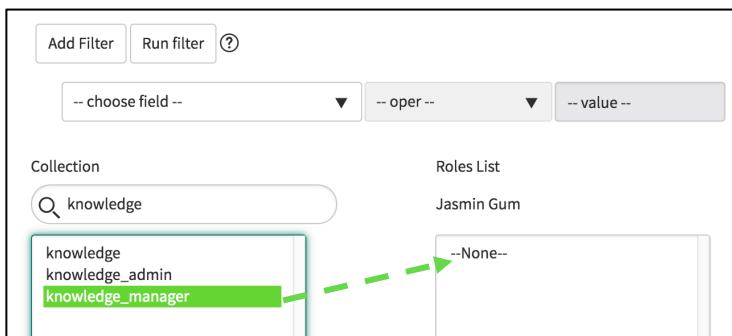
1. As the system administrator, navigate to **User Administration > Users**
2. Locate **Jasmin Gum**, a member of the Human Resources group, and open the record.

Name	User ID	Email
Jasmin Gum	jasmin.gum	jasmin.gum@cloudd.com

3. In the **Roles** tab (under Related Lists), click **Edit...**



4. Using the slushbucket, move the **knowledge manager** role to the Roles List by double-clicking on the role.



5. Select **Save**

## B. Create a new IT Knowledge Base Article

1. Impersonate **Jasmin Gum**
2. Navigate to **Knowledge > Import Articles**
3. Fill out the form as follows:
  - 1) **Knowledge Base:** *Human Resources*
  - 2) **Category:** *[leave empty]*
  - 3) Select the **Browse Files** button and browse your computer for the **Infinity\_Open\_Enrollment.docx** file
  - 4) **Double-click** the file name or **highlight** the file name and select **Open**

5) Select the **Import** button

\* Knowledge Base

1 Human Resources X ▾

Category

2 Select an option... ▾

\* Import a Word File

Drag and drop DOC or DOCX files here  
or

3 Browse Files

Article 1

Infinity\_Open\_Enrollment

Imported Document: Infinity\_Open\_Enrollment.DOCX

Import 5

4. Click **Continue** in the **Uploading Knowledge Articles** dialog box.

**Note:** Multiple files can be imported when you drag and drop multiple files into the **Import a Word File** box.

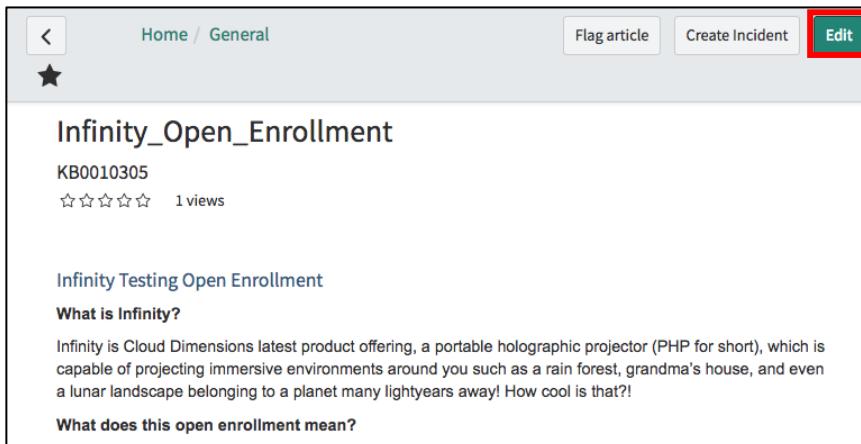
The **Short Description** is the same as the file name of the imported Word document.

5. Select the **knowledge article link** in the **Import Completed** dialog box.

Import Completed		
#	Record Number	Short Description
1	KB0010303	Infinity_Open_Enrollment

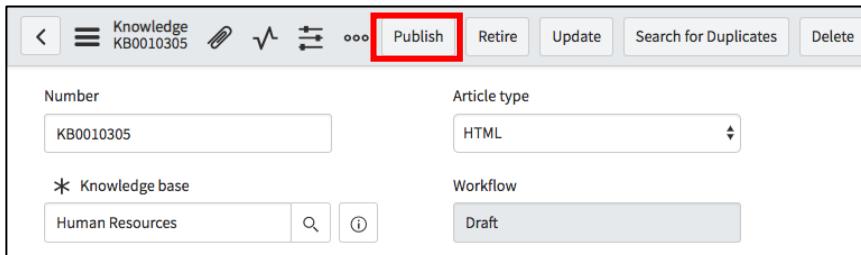
**Note:** The record number may be different than what is shown. When selecting the knowledge base article, it opens in a new tab.

6. Select the **Edit** button from the header



A screenshot of a ServiceNow Knowledge Base article titled "Infinity\_Open\_Enrollment". The article number is KB0010305 and it has 1 view. The content includes a section titled "Infinity Testing Open Enrollment" and a question "What is Infinity?". A note states: "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 house, and even a lunar landscape belonging to a planet many lightyears away! How cool is that?!" Below this is another question: "What does this open enrollment mean?". At the top right of the article view, there is a green "Edit" button with a red box drawn around it.

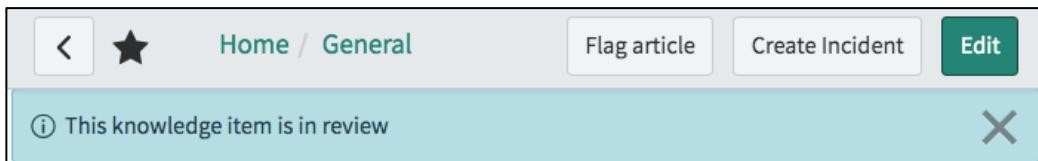
7. Select the **Publish** button



A screenshot of the "Publish" dialog box for article KB0010305. The "Number" field contains KB0010305. The "Article type" is set to "HTML". The "Knowledge base" is "Human Resources". The "Workflow" is "Draft". The "Publish" button at the top is highlighted with a red box.

**Note:** The default publish workflow for the Human Resources Knowledge Base is **Knowledge – Approval Publish**. This means after an author selects 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.

8. Verify the information message is displayed stating *This knowledge item is in review*



A screenshot of the ServiceNow Knowledge Base article view. At the top, the "Edit" button is visible. A blue banner at the bottom of the screen displays the message "This knowledge item is in review" with an information icon. There is also a close button (X) on the right side of the banner.

9. **Close** the current tab.

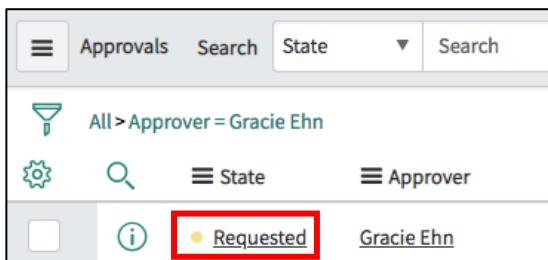
10. **Close** the *Import Completed* dialog box

## C. Approve the Article for Publishing

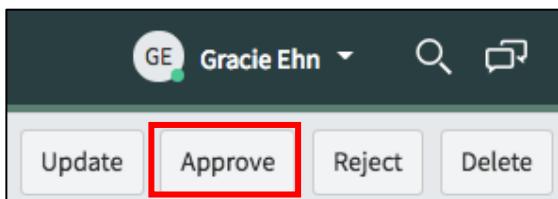
1. Impersonate **Gracie Ehn**, manager of Jasmin Gum and Knowledge Manager for the Human Resources Knowledge Base.

**Note:** Gracie Ehn is also a member of Human Resources and acts as the approver for new content submitted to be published.

2. Navigate to Service Desk > My Approvals
3. Locate the requested approval record and click the **Requested** link:



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:



**Note:** The information message, Approved Knowledge: KB0010... is displayed and the State of the Approval record is **Approved**.

6. Navigate to Self-Service > Knowledge
7. Open the **Human Resources** Knowledge Base to confirm the article appears:

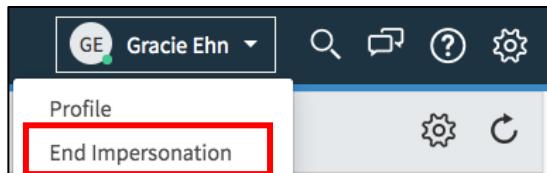
A screenshot of a web-based application interface titled "Human Resources". At the top, there are tabs for "All", "Articles", and "Questions", with "Articles" currently selected. Below the tabs is a search bar with a magnifying glass icon. The main content area displays a single article card. The article title is "Infinity\_Open\_Enrollment". Below the title, it says "Authored by System Administrator • 3 Views • Last updated 1m ago • Rating ★★★★★". The article content is summarized as: "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...". At the bottom of the card, it says "Knowledge Base: Human Resources".

## D. Create and Apply User Criteria

The Human Resources Knowledge Base is currently public to all users who 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. As Gracie Ehn, select **End Impersonation**



2. As the **System Administrator**, navigate to **Knowledge > Administration > User Criteria**
3. Select **New**
4. Fill in the form as follows:

**Name:** Cloud Dimensions Employees

**Companies:** Cloud Dimensions

**Note:** To select Cloud Dimensions, click on the **Unlock Companies** (padlock) icon, type cloud, select **Cloud Dimensions**, and click on the **Lock Companies** (open padlock) icon.

A screenshot of the 'User Criteria' creation form. At the top left is a back arrow and a 'User Criteria' title. To the right are buttons for 'New record', 'Edit', 'Delete', 'List', and 'Submit'. A note at the top says 'User Criteria may be used to restrict access to records in Service Catalog and Knowledge'. The form has fields for 'Name' (set to 'Cloud Dimensions Employees'), 'Application' (set to 'Global'), and 'Active' (checkbox checked). There are sections for 'Users' (with a lock and person icon), 'Groups' (with a lock icon), 'Companies' (with a lock icon next to 'Cloud Dimensions'), and 'Locations' (with a lock icon).

5. Select **Submit** to create the Cloud Dimensions Employees User Criteria record
6. Navigate to **Knowledge > Administration > Knowledge Bases**
7. Locate and open the **Human Resources** record

8. Scroll down and select the **Can Read** tab:

A screenshot of the ServiceNow interface. At the top, there are four tabs: 'Knowledge (3)', 'Questions', 'Can Read' (which is highlighted with a red box), and 'Can Contribute'. Below the tabs is a toolbar with icons for 'Can Read' (highlighted with a red box), 'New', 'Edit...', 'Search', and a dropdown menu labeled 'Can Read ▾'. Underneath the toolbar, there are two sections: 'Knowledge Base = Human Resources' and 'Can Read'. The 'Can Read' section contains a search bar and a list of items.

9. Select **Edit...** to add an *existing* User Criteria record to the Can Read list

10. Add **Cloud Dimensions Employees** to the **Can Read** List using the slushbucket

A screenshot of a 'Can Read List' dialog box. On the left, under 'Collection', there is a search bar and a list of criteria: "'Problem Analyzers' and 'Problem Solving' Group', 'All ACME Corporation employees', 'Users with 'Admin' Role', 'Users with 'cmdb\_read' Role', 'Users with 'knowledge' role', and 'Users with 'scrum\_master' or 'scrum\_admin' role''. To the right, under 'Can Read List', there is a list titled 'Human Resources' containing the item 'Cloud Dimensions Employees', which is highlighted with a red box. At the bottom of the dialog are 'Cancel' and 'Save' buttons.

11. Select **Save**

## Lab Verification

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. Navigate to **Self-Service > Knowledge**
3. Verify the **Human Resources** Knowledge Base is not visible.
4. Impersonate **Megan Burke**, an employee of Cloud Dimensions.
5. Verify the **Human Resources** Knowledge Base is visible.
6. Select the **1 Article** in the Human Resources Knowledge Base.

The screenshot shows the 'Knowledge Bases' list. A specific category, 'Human Resources', is highlighted with a blue border. Inside this box, the text '0 Questions and 1 Articles' is displayed, with '1 Articles' being specifically highlighted by a red rectangle.

7. Open the **Infinity\_Open\_Enrollment** knowledge article

The screenshot shows the details page for the knowledge article 'Infinity\_Open\_Enrollment'. At the top, there's a navigation bar with a back arrow, a star icon, 'Home / General', and buttons for 'Flag article', 'Create Incident', and 'Edit'. Below the title 'Infinity\_Open\_Enrollment' and ID 'KB0010302', there's a rating section with five stars and '3 views'. The main content area contains the article's text: 'Infinity Testing Open Enrollment' and 'What is Infinity?'. The text describes the product as a portable holographic projector capable of projecting immersive environments.

8. From the **User Menu**, **End Impersonation**

**Add another feather to your cap – you have learned how easy it is to import Word articles into a ServiceNow database!**

## Knowledge Check

What is the module used to import Word documents as articles?

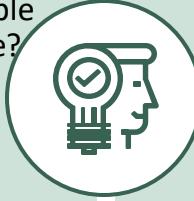
– **Knowledge > Articles > Import Articles**

What access does a user need to be able to import articles to a knowledge base?

– **Can contribute**

Which module is used to view knowledge content by category?

– **Self Service > Knowledge**



Which role can manage multiple knowledge bases?

– **knowledge\_admin**

What tool controls the publishing and retiring process for knowledge articles?

– **Workflows**

What is used to determine user access to knowledge bases or a knowledge article?

– **User Criteria**

## Section 4.2: Service Catalog

### User Story

---

As the **Infinity Product Owner**,

I want a **simple way for employees to order an Infinity device** through our service portal

So I can **track orders and inventory in one convenient location**.



## What is Service Catalog?

# Service Catalog

The **Service Catalog** is a robust ordering system for services and products offered by various departments for users:

- Categories organize catalog items
- “One stop shopping” offered to users
- Access to the Service Desk
- Help and Training Portal
- Multiple catalogs are supported

The screenshot shows a Service Catalog interface with a search bar at the top right. On the left, there's a sidebar titled "Top Requests" listing "Access", "Standard Laptop", "Apple iPad 3", "Cisco Jabber 10.5", and "Samsung Galaxy S7 Edge". Below the sidebar is a "Shopping Cart" section indicating it is empty. The main area is divided into several categories: "Services" (icon of wrench and screwdriver), "Hardware" (icon of computer monitor and smartphone), "Software" (icon of computer monitor and gear), "Office" (icon of office building), "Desktops" (icon of desktop computer), "Peripherals" (icon of computer monitor and keyboard), and "Mobiles" (icon of smartphone). Each category has a brief description.

The Service Catalog lets users see a list of things they might need (to create a request for) or would like to have – usually, but not limited to: IT products and services. Administrators and users with one of the various catalog roles 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.

The ServiceNow platform supports multiple Service Catalogs. Users with the **admin** or **catalog\_admin** role 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. This results in the ability to dynamically control the ordering options from user to user.

# Service Catalog Major Components

## Items



**Items** are the building blocks of the Service Catalog, including:

- Hardware
- Software
- Services

## Variables



**Variables** provide questions to help the requester specify what item, option, or service to order

## Variable Sets



**Variable Sets** are a modular unit of variables that can be shared between catalog items

## Record Producers



**Record Producers** are a form that produces a task record

## Order Guides



**Order Guides** assist customers in ordering a complete set of needed items and help users identify the item relationships

## Flows



**Flows** run behind the scenes and communicate the stages of the approval process to the requester, 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.

**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 individual variables to a catalog item, or multiple variables collected in a **Variable Set**.

**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.

**Order Guides:** Order Guides provide the ability to order multiple, related items as one request. Questions can be used to present item options, and present users with only **relevant questions and choices** at the appropriate time in the ordering process.

**Flows:** When you create a new service catalog item, you can create a new corresponding flow at the same time. This flow is used to drive complex fulfillment processes and can send notifications to defined users or groups.

# Service Catalog Items and Variables

To create a new item or modify an existing item, navigate to **Service Catalog > Catalog Definitions > Maintain Items**

Name	Short description	Active	Catalogs	Category	Price	Type	Updated
3M Privacy Filter - Lenovo X1 Carbon	Privacy Filter - X1 Carbon	true	Service Catalog	Peripherals	\$43.19	Item	2016-03-08 16:49:57
3M Privacy Filter - MacBook Pro	Privacy Filter	true	Service Catalog	Peripherals	\$42.23	Item	2016-03-08 16:49:48
3M Privacy Filter - MacBook Pro Retina	Privacy Filter	true	Service Catalog	Peripherals	\$40.31	Item	2016-03-08 16:49:59
Access	Microsoft Access	true	Service Catalog	Software	\$139.99	Item	2016-03-17 14:13:59
Acrobat	Adobe Acrobat	true	Service Catalog	Software	\$0.00	Item	2016-03-17 14:15:57
Add network switch to datacenter cabinet		true	Service Catalog	Network Standard Changes	\$0.00	Item	2015-07-01 07:07:37
Adobe Acrobat Pro	Create, edit or convert PDF files	true	Service Catalog	Software	\$0.00	Item	2016-03-08 16:49:48
Adobe Creative Cloud	More connected ways of creating and shar...	true	Service Catalog	Software	\$0.00	Item	2016-03-08 16:50:17

Once an item is published to the Service Catalog, users will be able to order it

## Variables

- Global by default
- Define the questions to ask the end user ordering the catalog item
- Question choices can define the available options and might affect the order price

### Examples:

- Which monitor size?
- Who is the hiring manager?
- What is the budget code?

Service Catalog variables are global by default and provide options to tailor a catalog item to the customer's needs. For example, a computer might be available with different operating systems.

The Service Catalog lets you attach variables either to a catalog item or to an execution plan.

### 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 which 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**. From the Application Navigator, 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.

Used by items and variables, the Order field establishes the sequence for displaying information. For example, an item with 100 in the Order field, displays first in the list. An item with 200 in the Order field will display second.

# Record Producers

**Record Producers** appear as simplified forms, allowing users to provide information that is translated into task-based records being added or modified in the database

Service Catalog > Can We Help You? > Create Incident

Create an incident record to report and request assistance with an issue you are having.

Request assistance with an issue you are having. An incident record will be created and managed through to successful resolution. You will also be notified of progress.

\* Urgency  
► More information  
1 - High

\* Please describe your issue below  
► More information  
I am unable to sign into the self-service portal on my mobile device.

Incident - INC0010001

This incident was opened on your behalf. The IT department will contact you if they need any further information. You can track status from this Homepage.

Number	INC0010001	Opened	2018-01-15 13:11:51
Caller	Joe Employee	Closed	
Watch list	Lock	Urgency	1 - High
	Watch	State	New

\* Short description I am unable to sign into the self-service portal on my mobile device.



- When employees use the Service Catalog on the HR Service Portal to submit a request for Direct Deposit setup, a record producer transfers the request into an HR case

- Incident management might use a record producer for users who need to open an incident easily via the web

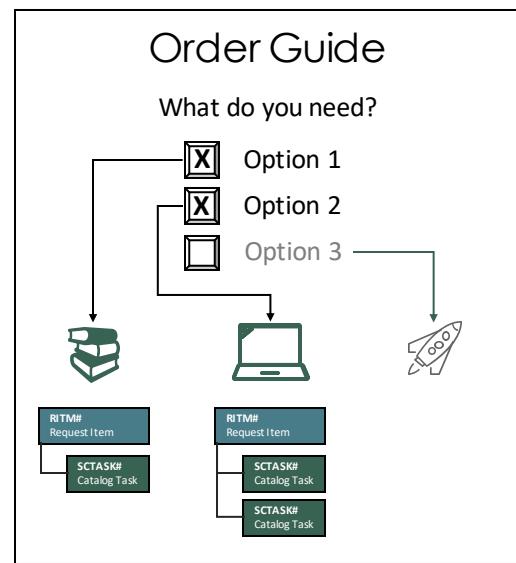
- Facilities management uses a record producer for users to open requests for printer service

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.

## Order Guides

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



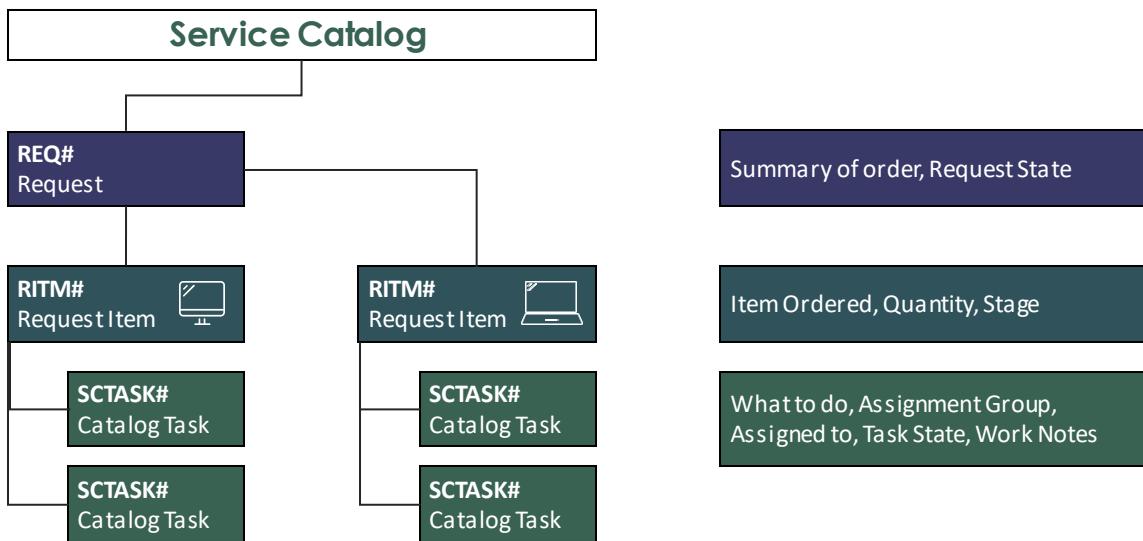
Items are the building blocks of the Service Catalog. Once you have built a complete item with variables and a delivery workflow, you can do a number of things with it, such as add it to an Order Guide.

Order Guides provide the ability to order multiple, related items as one request. Remember that variables are presented by the **Order** field number.

Use an Order Guide to assist users in determining what items they need.

# Service Catalog Item Request Output

now.



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.

In this example, a manager orders two new computers for the team.

The first requested item is a desktop computer with two catalog tasks associated to it: order equipment and configure equipment.

The second requested item is a laptop computer with two catalog tasks associated to it: order equipment and configure equipment.

Although the requested items share similarly named catalog tasks, these tasks are tied directly to their respective item because the required steps for completion might be different from item to

# Progress Stages for a Requested Item

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

The screenshot shows a ServiceNow request list interface. At the top, there are tabs for 'Requested Items (1)', 'Approvers (1)', 'Group approvals (1)', and 'Recurring Prices (1)'. Below the tabs is a search bar with dropdowns for 'Number' and 'Search'. The main area displays a single request record with the following details:

- Request ID:** REQ0010103
- Item:** RITM0010002
- Quantity:** 1
- Created:** 2018-01-20 13:08:06
- Price:** \$1,100.00

Below the record, there is a section titled 'Actions on selected rows...' with a dropdown arrow. To the right of the record, a 'Stage' column is expanded, showing the following options:

- Waiting for Approval (Waiting for Approval (In progress))**
- Dept. Head Approval - 2 Days (Dept. Head Approval - 2 Days (Pending - has not started))
- CIO Approval - 2 Days (CIO Approval - 2 Days (Pending - has not started))
- Order Fulfillment - 4 Days (Order Fulfillment - 4 Days (Pending - has not started))
- Backordered - 14 Days (Backordered - 14 Days (Pending - has not started))
- Deployment - 1 Day (Deployment - 1 Day (Pending - has not started))
- Completed (Completed (Pending - has not started))

After an request has been submitted, users are able to easily track it by navigating to **Self-Service > My Requests** and opening the record associated with the request.

Workflows attached to an item indicate the progress or state of an item in the delivery process with one of the following stages:

- **Waiting for approval (In Progress)**
- **Approved**
- **Pending (has not started)**
- **Fulfillment (In Progress)**
- **Deployment/Delivery**
- **Completed**

Additionally, workflows can have multiple rounds of approval actions as it relates to Service Catalog requests. Each approval action can share the same stage value or their own separate stages.

Stages can be grouped into a Stage Set for convenience of applying related stage values from workflow to workflow.

# Service Catalog Security: User Criteria

now.

**User Criteria** defines conditions that are evaluated against users to determine which users can access Service Catalog items.

You can apply several user criteria records to a single catalog item or category.

User Criteria is applied to an item or category.

The screenshot shows the 'User Criteria' record form. At the top, it says 'User Criteria' and 'Users with 'Admin' Role'. Below that is a table with columns for 'Name' (containing 'Users with ''Admin'' Role') and 'Criteria Type' (with options for 'Users', 'Groups', 'Roles', and 'Advanced'). Under 'Users', there are icons for a user and a group. Under 'Roles', there is an icon for a role and the text 'admin'. To the right of the table are sections for 'Application' (set to 'Global'), 'Active' (checkbox checked), and 'Companies', 'Locations', 'Departments' (each with an icon and a lock symbol). There is also a 'Match All' checkbox. 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.



**Time**  
**10-15m**

- Create an Infinity service catalog item
- Add item variables
- Validate functionality with "Try It" option

## Lab 4.2: Create a Catalog Item

As the **Infinity Product Owner**,

I want a **simple way for employees to order an Infinity device** through our service portal

So I can **track orders and inventory in one convenient location**.

# Create a Catalog Item

Lab

4.2

10-15m

## Lab Objectives

You will achieve the following objectives:

- Create an Infinity service catalog item
- Add item variables
- Validate your work with “Try It”

## Scenario

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.

**Required Resource(s):** CloudDimensions-Infinity-Logo.png

## A. Create New Service Catalog Item

1. From the *User Menu*, impersonate **Asset Manager**

**Note:** This user account is provided the **catalog\_admin** role, which grants them access to the **Maintain Items** module in order to complete Service Catalog administration tasks.

2. Navigate to **Service Catalog > Catalog Definitions > Maintain Items**
3. Select **New**

4. Fill out the top of the form as shown:

Name: **Infinity**

5. Click the **lock** icon (Unlock Catalogs) next to Catalogs

The screenshot shows the 'Catalog Item' form with a 'Name' field containing 'Infinity'. Below it is a 'Catalogs' field with a lock icon, which is highlighted with a red box. A tooltip below the field says, 'If you want users to be able to search for this item, add it to a Category'.

6. Start typing **ser** then select **Service Catalog**

The screenshot shows a 'Catalogs' selection dialog. A search bar at the bottom has 'ser' typed into it. Below the search bar is a list box containing 'Service Catalog', which is highlighted with a green border.

7. Select the **lock** icon (Lock Catalogs) to close the selection box.

8. In the Category field, select **Hardware**

The screenshot shows the 'Catalog Item' form again. The 'Name' field is 'Infinity'. The 'Catalogs' field is set to 'Service Catalog' with a lock icon. The 'Category' field is set to 'Hardware'. Other fields like 'Active' (checked), 'Availability' (Desktop Only), and a toolbar button are also visible.

**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 hierarchically.

9. 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!**

10. Select the **Pricing** tab and add in the **Price** field, enter **395.99**

The screenshot shows the Service Catalog item creation interface. The 'Name' field is set to 'Infinity'. The 'Active' checkbox is checked. Under 'Catalogs', 'Service Catalog' is selected. In the 'Category' section, 'Hardware' is chosen. The 'Pricing' tab is currently active. The 'Price' field is highlighted with a red box and contains '\$ 395.99'. Below it, there are fields for 'Recurring price' (set to \$ 0.00) and 'Recurring price frequency' (set to '-- None --').

11. Select the **Picture** tab and in the **Picture** field, select **Click to add...**

The screenshot shows the Service Catalog item creation interface. The 'Category' section shows 'Hardware'. The 'Picture' tab is selected. Below it, there are fields for 'Icon' (labeled 'Click to add...') and 'Picture' (also labeled 'Click to add...', which is highlighted with a red box).

12. Choose File: **CloudDimensions-Infinity-Logo.png**

13. Select **OK**.

14. Select **Update**.

## B. Verify the Service Catalog Item Creation

1. Navigate to **Self-Service > Service Catalog**
2. Select the **Hardware** category header.
3. Select the **Infinity** item from the list to open the order screen.

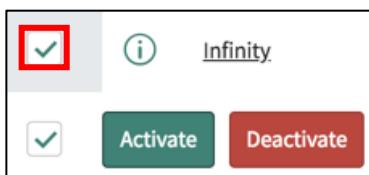
## C. Add a Memory Variable to the Service Catalog Item

From this form, a user could order an Infinity, but the form does not yet allow users to specify options for the device. Let us fix that!

1. Navigate to **Service Catalog > Catalog Definitions > Maintain Items**
2. Filter the list by searching for **Name \*infinity**

The screenshot shows a list of catalog items. At the top, there is a search bar with the text '\*infinity'. Below the search bar, there are several filters: 'Name' (set to '\*infinity'), 'Active' (set to 'true'), 'Catalogs' (set to 'Service Catalog'), and 'Category' (set to 'Hardware'). At the bottom of the list, there is one item named 'Infinity' with a checked checkbox. There are 'Activate' and 'Deactivate' buttons at the bottom.

3. Select the checkbox to the left of the **Infinity** record.



4. Select the **Deactivate** button.

**Note:** This button deactivates the catalog item(s) selected on the list, making them inaccessible to users in the Service Catalog. Until we finish defining our item, keep it deactivated.

5. Open the **Infinity** record.
6. From the **Variables** tab, select **New**

The screenshot shows the 'Variables' tab. A red box highlights the 'New' button. Below the tab, there is a search bar with the text 'Catalog item = Infinity'.

7. Fill out the following fields on the form:

**Type:** Multiple Choice  
**Mandatory:** [select checkbox]

8. Under the **Question\*** tab, enter the Memory question as shown:

Question: **How much memory do you want in your Infinity?**

Name: **memory**

**Note:** The Name auto-fills with *how\_much\_memory\_do\_you\_want\_in\_your\_infinity*. Replace this with the name, **memory**.

*The corporate-approved version of the Infinity is available in both 256GB and 512GB models.*

9. Select the **Default Value\*** tab.

10. Enter the Default value: **256**

11. **Save** the form, instead of Submit, to define the memory Question Choices.

12. Scroll to the **Question Choices** section then select the **New** button.

The screenshot shows a user interface for managing question choices. At the top, there's a toolbar with icons for 'Question Choices' (a list icon), 'New' (highlighted with a red box), 'Search' (two search icons), 'Order' (down arrow), and another 'Search' button. Below the toolbar, there's a list area with a green filter icon and the text 'Question = How much memory do you want in your Infinity?'. Underneath this list are three buttons: a gear icon labeled 'Text', a magnifying glass icon labeled 'Value', and another gear icon.

13. Fill out the **Question Choices** form as shown:

Text: **256GB**

Value: **256**

14. **Submit**

15. In the **Question Choices** section, click **New**.

16. Fill out the form for the second memory choice:

Price: **\$100.00**

Text: **512GB**

Value: **512**

**Note:** The Price field is utilized by 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.

17. **Submit**

18. You should now see both question choice values:

Question = How much memory do you want in your Infinity?		
	Text	Value
	<u>256GB</u>	256
	<u>512GB</u>	512

19. Select **Update** to return to the Infinity item record.

## D. Add Color Variable to Service Catalog Item

With the first variable defined to provide end users with a choice of memory options, you will now create a second question for choosing a color choice by configuring a second variable.

1. In the **Variables** related list, select **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** the record (do not Submit)
4. Scroll to the **Question Choices** section and select 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 Context 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.

Visit [docs.servicenow.com](https://docs.servicenow.com) for the definitions of all the possible variable types.

9. Select **Update** to return to the Infinity item record.

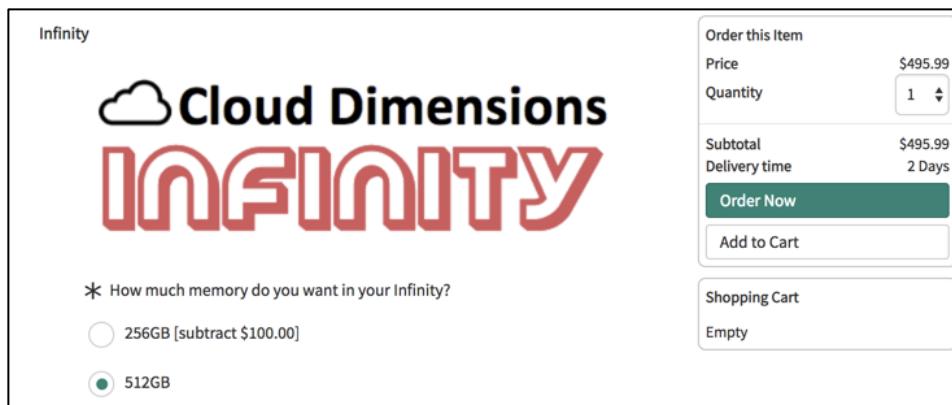
## Lab Verification

### Test the Catalog Item

1. From the **Infinity** Catalog Item form, check the **Active** field to activate the item.
2. **Save** the record
3. Next, click the **Try It** button from the form header to view the item order screen, with the new variable options added.

**Note:** The Try It button is only available if the item is active.

4. Choose **512 GB** and notice how the item **Price** changes from \$395.99 to \$495.99.



5. From the **User Menu**, **End Impersonation**

**Congratulations! Now that you have created the Infinity in the Service Catalog, you're ready to create the flow to fulfill a new Infinity request!!**

## Knowledge Check

What is created when an order is placed for a catalog item?

- **REQ# (Request)**
- **RITM# (Requested Item)**
- **SCTASK# (Service Catalog Task)**

Which module is used to add, update, or remove catalog items?

- **Maintain Items**

What is attached to a catalog item and determines the fulfillment process?

- **Workflow or Flow Designer Flow**



What represents questions asked when ordering a catalog item?

- **Variables**

What are some examples of variables?

What represents multiple related items grouped as one request?

- **Order Guide**

What appears as a simplified form to create a record in a table?

- **Record Producer**

## Section 4.3: Flow Designer

### User Story

---

As the **Asset Manager**,

I want **to automate the procurement process** for Infinity devices

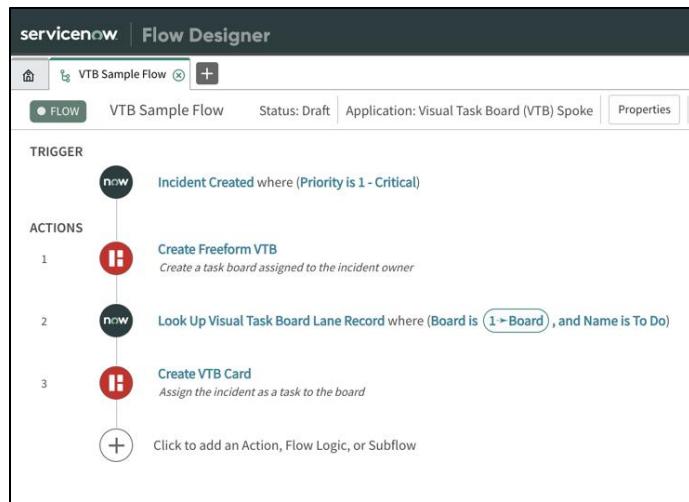
So employees can receive their devices in a timely manner and **we can improve our team's CSAT scores**.



# What is the Flow Designer?

The **Flow Designer**, **Flow Designer > Designer**, is a non-technical interface for building and enabling process automation capabilities, known as **flows**.

Flows automate business logic for a particular application or process such as approvals, tasks, notifications, and record operations.



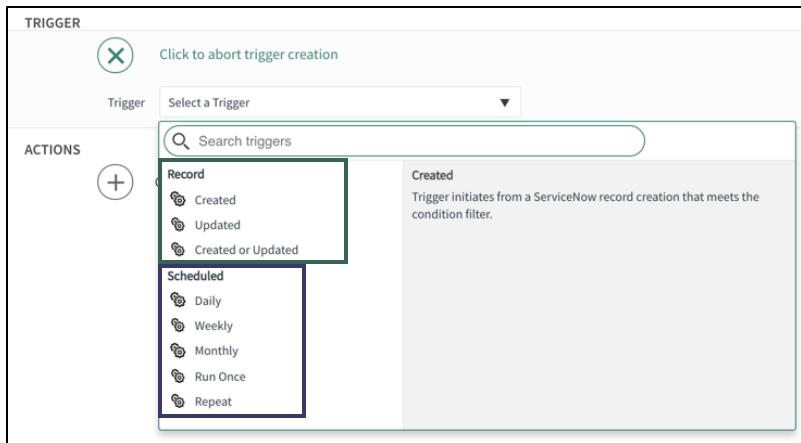
Launch the Flow Designer by navigating to **Flow Designer > Designer** in the Application Navigator. The following roles might be required to work with the Flow Designer in one capacity or another: **flow\_designer**, **flow\_operator**, and **action\_designer**.

A flow consists of a *trigger* and one or more *actions*.

NOTE: You can expand the Flow Designer solution to integrate with external instances and third-party applications with a separate subscription to [IntegrationHub](#).

# Flow Components: Triggers

now.



Triggers can be *record-based*, *schedule-based*, or *application-based*.

- **Record-based** triggers run a flow after a record has been created, updated, or deleted.
- **Schedule-based** triggers run a flow at the specified date and time.

Triggers instantiate the flow and can be *record-based*, *schedule-based*, or *application-based*.

- **Record-based** triggers run a flow after a record has been created, updated, or deleted. When using a record-based trigger, the triggering record can be used later in the flow as input for actions.
- **Schedule-based** triggers run a flow at the specified date and time: daily, weekly, monthly, etc. The execution time can be used as an input for actions in the flow.
- **Application-based** triggers are added when the associated application spoke\* is activated. In some instances, a plug-in might need to be activated as well (e.g. Flow Designer Support for Service Catalog).

\*A *spoke* contains Flow Designer triggers and actions dedicated to a particular application. For example, the **ITSM Spoke** contains actions for managing Task records such as the **Create Task** action. Spokes are activated when their parent application is activated.

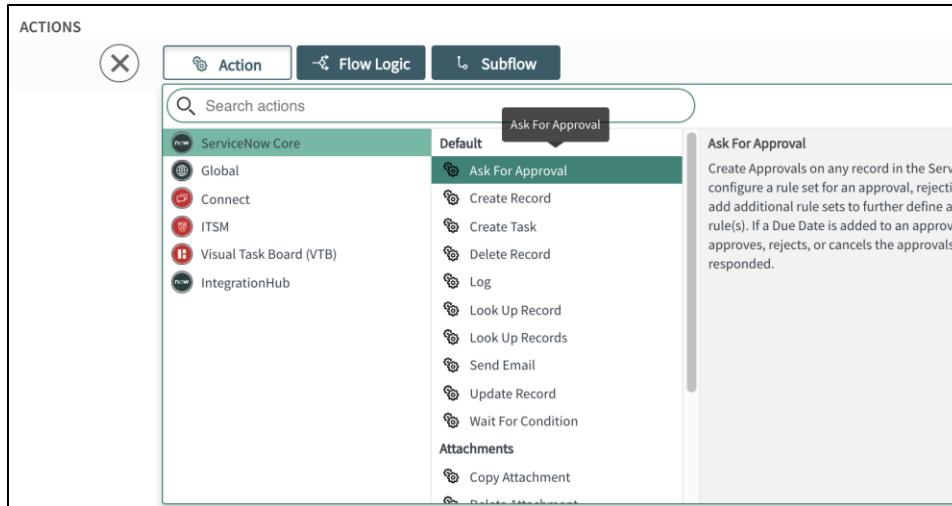
# Flow Components: Trigger and Condition

Specific conditions can be set to identify when a flow should be instantiated.

The screenshot shows the ServiceNow Flow Designer interface for the "VTB Sample Flow". The flow is currently in Draft status and is associated with the "Visual Task Board (VTB) Spoke" application. The "Trigger" section is selected, displaying the "Incident Created" trigger. The trigger is set to "Created" and is linked to the "Incident [incident]" table. A condition is defined: "Priority is 1 - Critical". The condition is set to "All of these conditions must be met". There are buttons for "Delete", "Cancel", and "Done" at the bottom right.

## Flow Components: Actions

**Actions** are operations executed by the system, such as looking up a record, updating a field value, requesting an approval, or logging a value.



ServiceNow Core actions include:

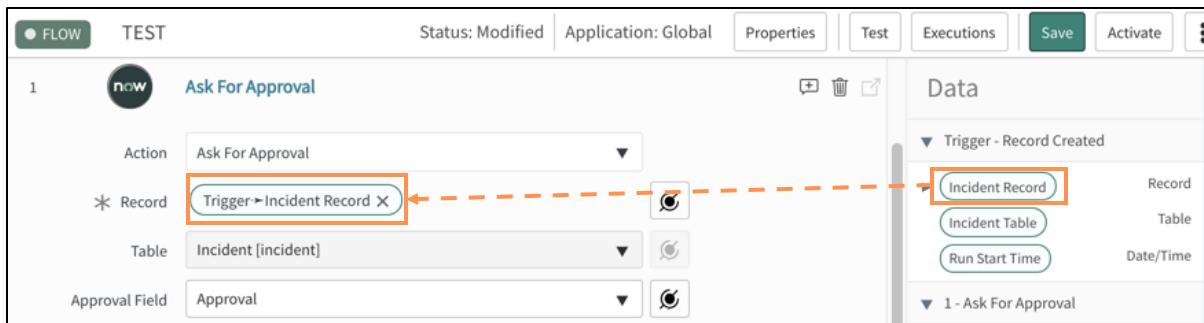
- **Ask for Approval** – create approvals on any record, including rules for an approval, rejection, or cancellation, and a due date
- **Create Record** – create a record on any table with configurable fields and field values
- **Delete Record** - delete a record on any table
- **Look Up Record** – look up a single record on any table, confirming whether or not it exists and using the information from the record in subsequent actions within the flow.
- **Wait for Condition** - Pause the flow until the record conditions are met. For example, wait for the State to change to Pending.

In addition to these core actions, new application-specific core actions can be created by activating the associated spoke.

## Flow Components: Data

Each time you add an action to a flow, Flow Designer adds a *data pill* to store its results.

The **Data** section of the Flow Designer contains data pills that can be used in subsequent actions. To reference the data stored in the data pill, drag and drop the data pill from the Data section to the appropriate field in the flow or click on the Data Pill Picker icon (  ).



When designing a flow, you can use the results of an action as inputs for other flows, actions, or subflows. Using the sequence value (Trigger, 1, 2, 3, etc.), you can ensure you are using the correct data pill as an input value.

When a flow runs an action, it generates the data pill runtime value, which remains the same for the duration of the flow. For example, a data pill for **[Trigger->Incident record]** always contains the incident record values from when the flow *started*.

**NOTE:** When using the **Data Pill Picker** icon, use the arrow keys to dot-walk to fields in other tables.

# When to Use Flows



## Use

To orchestrate business processes across services with little technical user knowledge

To reduce technical debt; reduced script and script reuse to simplify upgrades and deployments

When integrating with 3<sup>rd</sup> party systems



## Do Not Use

When heavy scripting is required to provide the majority of automation and execution

Existing logic already developed using Workflow

Instance is running Jakarta or prior



## Real World Use Cases

- Project manager has tasks automatically created and added to a Visual Task Board when a specific record is created

- Customer Service develops a Flow to communicate incident resolution through the end-user support channel

- Change Manager creates an outage record and links it to an incident when the primary email server goes offline

## Other benefits of using flows:

- Single environment to build and visualize business processes
- Configuration and runtime information available to create, operate, and troubleshoot flows from a single interface
- Provides natural-language-descriptions of flow logic
- Promotes process automation by enabling subject matter experts to develop and share reusable actions
- Allows extending Flow Designer content by subscribing to IntegrationHub or installing spokes



**Time**  
**20-25m**

- Create a new flow from scratch
- Associate the flow to the Infinity Service Catalog item
- Test the flow

## Lab 4.3: Create a Flow Designer Flow

As the **Asset Manager**,

I want to **automate the procurement process** for Infinity devices

So employees can receive their devices in a timely manner and **we can improve our team's CSAT scores**.

# Lab

## 4.3

⌚20-25m

# Create a Flow Designer Flow

## Lab Objectives

You will achieve the following objectives:

- Create a new flow from scratch
- Associate the flow to a Service Catalog item
- Test the flow

**Lab Dependency:** Requires the completion of Lab 4.2: Create a Catalog Item

## Scenario

Now that Cloud Dimensions has opened 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 Flow.

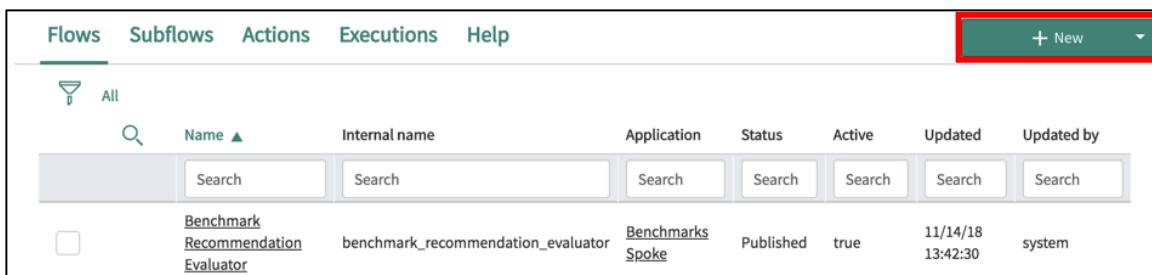
**Note:** The *Flow Designer Support for Service Catalog* plugin has been activated in your student instance. It is not installed by default but is necessary to create flows for service catalog.

## A. Create a Flow Designer flow from scratch

1. As **System Administrator**, navigate to **Flow Designer > Designer**

**Note:** The Flow Designer opens and displays in a separate tab or browser window.

2. Click the **+ New** button:



Flows								
Subflows								
Actions								
Executions								
Help								
<b>+ New</b>								
All	Name ▲	Internal name	Application	Status	Active	Updated	Updated by	
<input type="checkbox"/>	<a href="#">Search</a>	<input type="text"/> Search	<input type="button"/> Search	<input type="button"/> Search	<input type="button"/> Search	<input type="button"/> Search	<input type="button"/> Search	
<a href="#">Benchmark Recommendation Evaluator</a>		benchmark_recommendation_evaluator	Benchmarks Spoke	Published	true	11/14/18 13:42:30	system	

3. Next, choose **Flow**

The screenshot shows the ServiceNow Flows list interface. At the top, there are tabs for Flows, Subflows, Actions, Executions, and Help. In the top right corner, there is a green button labeled '+ New' with a dropdown arrow, and a red box highlights the 'Flow' option in the dropdown menu. Below the tabs, there is a search bar and a table with columns: Name, Internal name, Application, Status, Active, Updated, and Action. The table contains three rows: Benchmark, Recommendation, and Evaluator.

4. Complete the form as follows:

**Name:** *Infinity Item Request*

**Run As:** *System User*

5. Select **Submit**

6. Verify the Flow Designer interface appears:

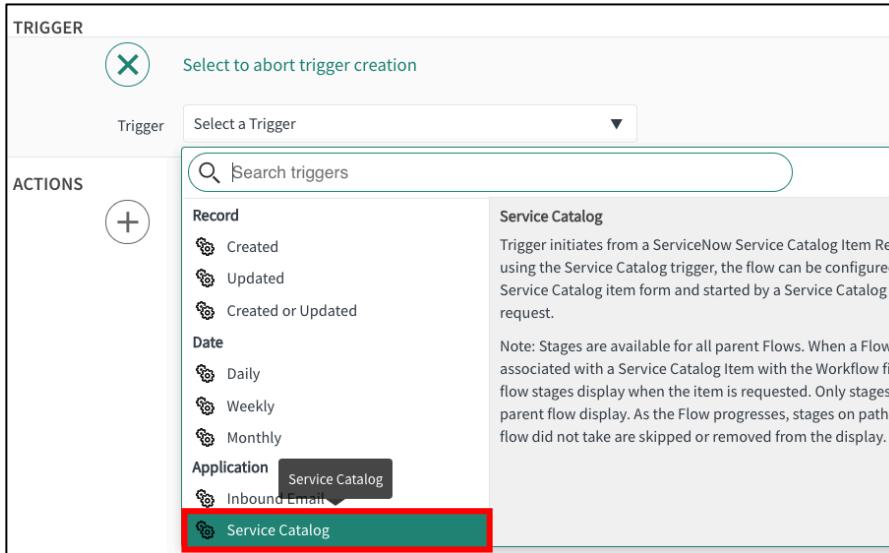
The screenshot shows the ServiceNow Flow Designer interface for the 'Infinity Item Request' flow. The title bar says 'servicenow | Flow Designer'. The main area has tabs for Home, Flow (selected), and Properties. The Properties tab is active. Below it, there are sections for TRIGGER and ACTIONS. The TRIGGER section has a plus sign button and the text 'Click to add a Trigger'. The ACTIONS section also has a plus sign button and the text 'Click to add an Action, Flow Logic, or Subflow'.

## B. Define a Trigger

1. Select **Select to add a Trigger**:

The screenshot shows the ServiceNow Flow Designer interface for the 'Infinity Item Request' flow. The title bar says 'servicenow | Flow Designer'. The main area has tabs for Home, Flow (selected), and Properties. The Properties tab is active. Below it, there are sections for TRIGGER and ACTIONS. The TRIGGER section has a plus sign button with the text 'Select to add a Trigger', which is highlighted with a red box.

- Choose **Service Catalog** from the Application section:

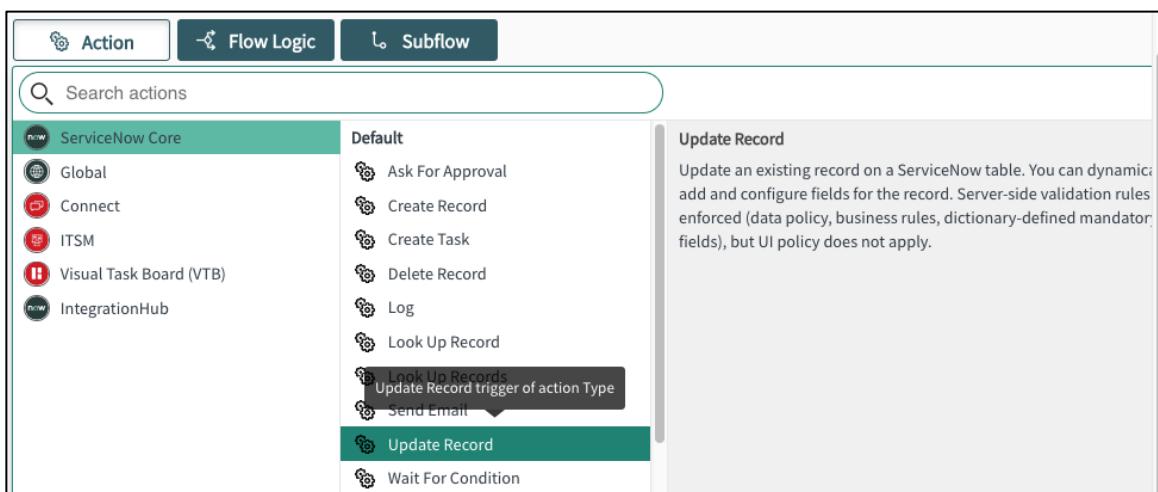


## C. Add an Action to Update a Requested Item Record

- Select the plus (+) icon to the left of **Select to add an Action, Flow Logic, or Subflow**
- Select **Action**



- Select the **Update Record** default action type under ServiceNow Core.



4. Choose the Requested Item record to update:
  - a) In the **Data Panel**, expand the **Trigger - Service Catalog** section.
  - b) Drag the **Requested Item Record** pill from the Data Panel to the **Record** field.
  - c) Release your mouse to “drop” the pill into the **Record** field.

The screenshot shows the 'TRIGGER' section with a 'now' trigger. Under 'ACTIONS', there is one step: '1 now Update Record'. The 'Action' dropdown is set to 'Update Record'. The 'Record' field has a dropdown menu open, showing 'Drag and drop record data pill' and 'Requested Item Record'. The 'Requested Item Record' pill is highlighted with a red box. The 'Table' field has a dropdown menu open, showing 'Select a Table'. To the right, the 'Data' panel is expanded to show the 'Trigger - Service Catalog' section, which contains the 'Requested Item Record' pill. Below it, the '1 - Update Record' section also lists the 'Requested Item Record' pill under the 'Record' field.

**Note:** This will automatically populate the Table field with Requested Item [sc\_req\_item].

5. Select the **+ Add Field Value** button:



6. Set **State** to *Pending*

The screenshot shows the 'Update Record' configuration screen. It includes sections for 'Action' (set to 'Update Record'), 'Record' (set to 'Trigger->Requested Item Record'), 'Table' (set to 'Requested Item [sc\_req\_item]'), and 'Fields' (set to 'State' with value 'Pending'). At the bottom left of the configuration area, there is a button labeled '+ Add Field Value'.

7. Select **Done**.

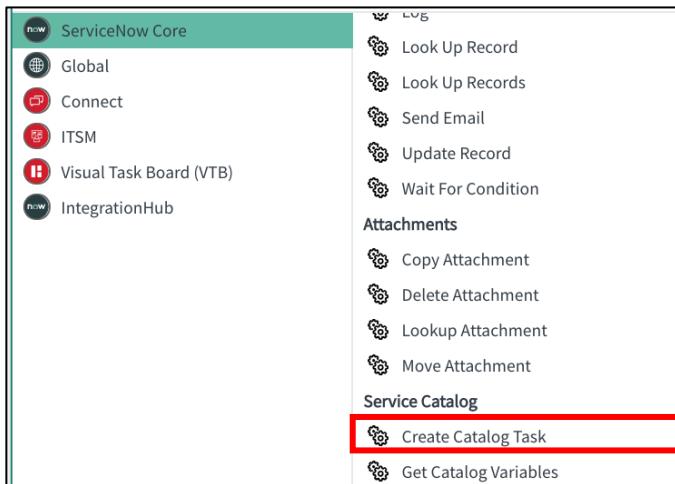
8. **Save** the changes you have made to the flow.



## D. Add a Create Catalog Task Action

1. Select the plus (+) icon to the left of **Select to add an Action, Flow Logic, or Subflow**
2. Select **Action**

3. Select the **Create Catalog Task** action within ServiceNow Core > Service Catalog:



4. Drag and drop the **Requested Item Record** pill from section 1 - Update Record to the **Requested Item [Requested Item]** field.
5. Update the **Short Description** field to **Infinity Delivery**
6. For **Fields [Catalog Task]**, select **+ Add Field Value** and add the following:

**Assignment Group | Service Desk**  
**State | Open**

Action	Create Catalog Task
Table Name	Catalog Task [sc_task]
* Requested Item [Requested Item]	1->Requested Item Record X
Short Description	Infinity Delivery
Fields [Catalog Task]	Assignment group State

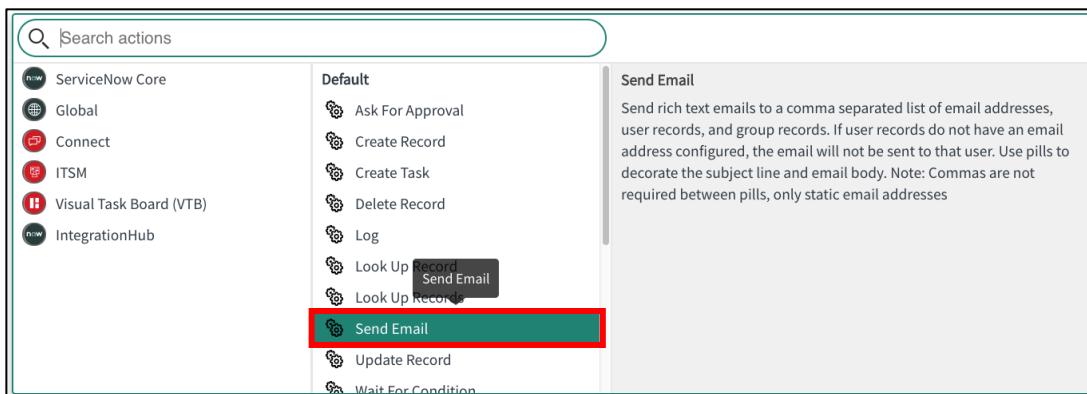
**Note:** Notice the **Wait** field (not shown above) is checked by default. This pauses the flow until the task completes and is no longer active (e.g. when the task is marked Closed Complete)

7. Select **Done**

## E. Send an Email to the Requester

Once the task is marked Closed Complete, the flow will resume, and an email confirmation will need to be sent to the requester.

1. Select the plus (+) icon to the left of **Select to add an Action...**
2. Select **Action**
3. Under **ServiceNow Core > Default**, locate and select the **Send Email** action type.



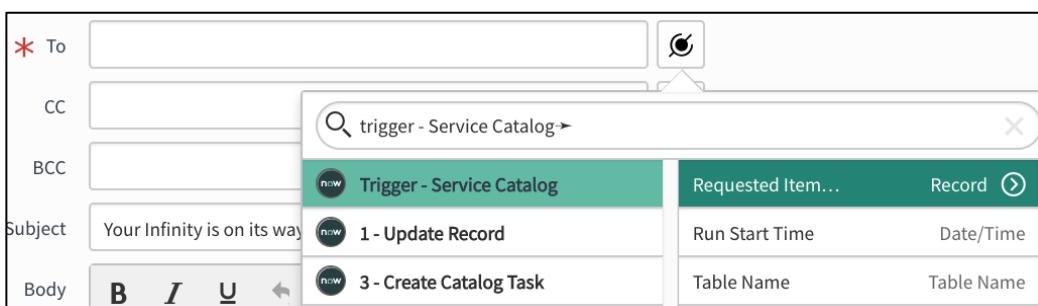
**Note:** We will use dot-walking to locate the email address of the requester, so we can send a confirmation when their order has been fulfilled.

4. Select the **Data Pill Picker** icon at the right of the **To** field.



**Note:** Selecting the Data Pill Picker icon will open a search box where you can dot-walk to select the email address of the requester for the requested item.

5. With **Trigger - Service Catalog** highlighted, press the **right arrow key** on your keyboard to highlight **Requested Item...**



6. Press the **right arrow key** again to dot-walk to the Requested Item table.
7. **Arrow down to Request**, then press the **right arrow key** to dot-walk to the Request table.

**Note:** Rather than arrow down to a specific field, you can start typing the name of the field and then press the right-arrow key to dot-walk to the next table.

8. **Arrow down to Requested for**, then press the **right arrow key** to dot-walk to the User table.
9. Finally, **arrow down to Email** and press the **Enter key** on your keyboard.

Trigger - Service Catalog		Requested Item...	Record >	Price	Currency	Order	Integer	Email	Email
Trigger - Service Catalog		Run Start Time	Date/Time	Priority	Choice	Parent	Reference		
1 - Update Record		Table Name	Table Name	Quantity	Integer	Price	Currency		
3 - Create Catalog Task				Reassignment co...	Integer	Priority	Choice		
				Recurring Price	Price	Reassignment co...	Integer		
				Recurring price fr...	Choice	Request state	Choice		
				Request	Reference >	Requested for	Reference >		

10. Complete the rest of the action as follows:

**Subject:** Your Infinity is on its way!

**Body:** Greetings!

Get ready to be immersed in new worlds with your Infinity Portable Holographic Projector (PHP)!! Your requested item has been shipped and is expected to arrive soon.

**Challenge:** Use the **Data Pill Picker icon** to add fields to the email body!

Subject: Your Infinity is on its way!

Body:

Greetings Trigger->Requested Item Record->Request->Requested for->First name!

Get ready to be immersed in new worlds with your Infinity Portable Holographic Projector (PHP)!!

Your requested item, Trigger->Requested Item Record->Item->Name has been shipped and is expected to arrive soon!

11. Select **Done**

## F. Add an Action to Update the Requested Item Record

Now that the catalog task is complete and the email notification has been sent, the State of the **Requested Item** needs to be updated to **Closed Complete**.

Remember the Request Management hierarchy:  
**Request** -> **Request Item** -> **Catalog Task**

1. Select the plus (+) icon to the left of **Select to add an Action, Flow Logic, or Subflow**.
2. Select **Action**
3. Select the **Update Record** default action type under ServiceNow Core.
4. Choose the **Requested Item** record to update:
  - a) In the **Data Panel**, expand the **1 – Update Record** section.
  - b) Drag the **Requested Item Record** pill from the Data Panel to the **Record** field.
  - c) Release your mouse to “drop” the pill into the **Record** field.

**Note:** This will automatically populate the Table field with **Requested Item [sc\_req\_item]**.

5. Select the **+ Add Field Value** button:



6. Set **State** to **Closed Complete**
7. Select **Done**
8. **Save** the changes you have made to the flow.



## G. Add an Action to Update the Request Record

The next step is to update the **Request State** for the **Request** to **Closed Complete**.

1. Select the plus (+) icon to the left of **Select to add an Action, Flow Logic, or Subflow**.
2. Select **Action**

3. Select the **Update Record** default action type under ServiceNow Core.
4. Choose the **Request** record to update:
  - a) In the Data Panel, expand the **1 – Update Record** section, then expand the **Requested Item Record** data pill.
  - b) Drag and drop the **Request** pill from the Data Panel to the **Record** field.

**Note:** This will automatically populate the Table field with **Request [sc\_request]**.
5. Select the **+ Add Field Value** button:
6. Set **Request State** to *Closed Complete*
7. Select **Done**
8. **Save** the changes you have made to the flow.

## H. Activate and associate flow to Service Catalog Item

---

*At this point, the Infinity Item Request flow features one catalog task: to deliver the item to the requester.*

*Additional actions can be added to the flow to enhance the process and experience but for now, we will activate and associate the flow to the **Infinity** Service Catalog item.*

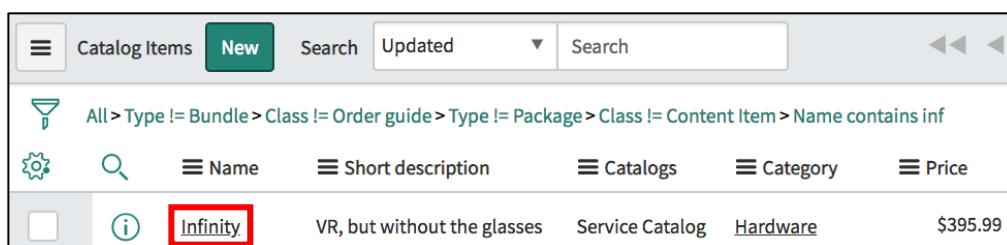
---

1. Click the **Activate** button:



**Note:** A message will appear confirming you would like to activate the flow – select **OK**.

2. Return to the main ServiceNow window and navigate to **Service Catalog > Catalog Definitions > Maintain Items**
3. Locate and open the **Infinity** item record:



4. Enter **Infinity Item Request** into the **Flow** field

The screenshot shows the 'Process Engine' tab selected in the top navigation bar. Below it, there are three input fields: 'Flow' (containing 'Infinity Item Request'), 'Workflow' (empty), and 'Execution Plan' (empty). Each field has a search icon (magnifying glass) and a help icon (info symbol) to its right.

**Note:** When there is a value in the Flow field of the Process Engine, the value in the Execution Plan field is removed.

5. Update the Infinity item record.

## LAB VERIFICATION

### Test the Flow – Order an Infinity

1. Impersonate **Joe Employee**.
2. **Self-Service > Service Catalog**
3. Select the **Hardware** Category, then select **Infinity**
4. You can request either **color**, and if desired, additional **memory**.
5. Select the **Order Now** button to make the request and initiate the flow.
6. Confirm an information message verifying the request **was submitted** is displayed at the top of the form.

### Test the Flow – Complete Delivery Tasks

We can review the progress of the Infinity Item Request flow by confirming the Item Delivery catalog task was created. Then, as a Service Desk member, close the task.

1. Impersonate **Beth Anglin**
2. **Service Desk > My Groups Work**
3. Locate and open the Service Catalog Task (SCTASK00100...) associated with the Delivery flow task. **HINT:** Search the Short description column for **delivery**.

4. Assume the item is in stock and can be delivered. Select **Close Task**

**Note:** This will mark the catalog task as Closed Complete for the Requested Item. It will also trigger the email notification to Joe Employee.

Test the Flow – Confirm email notification is in the Outbox

1. From the *User Menu*, **End Impersonation**
2. **System Mailboxes > Outbound > Outbox**
3. Confirm an email with subject “*Your Infinity is on its way!*” was created.

***Impressive!***

***You created your first Flow and associated it with a Service Catalog item!***

## Knowledge Check

What is the name of the interface for building process automation capabilities?

– **Flow Designer**

Which module is used to create or edit flow designer flows?

– **Flow Designer > Designer**

Where is the data from an action stored so it can be used in subsequent actions in the flow?

– **Data Pill**



When attaching a flow to a service catalog item, which plugin needs to be activated?

– **Flow Designer support for the Service Catalog**

What are the components of a Flow Designer flow?

– **Trigger**  
– **One or more Actions**

## Module 5: Introduction to Development

User Interface and Navigation

Collaboration

Database Administration

Self-Service and Automation

**Introduction to Development**

Capstone Project

### Module Objectives

- Demonstrate how to create a UI Policy and UI Policy Action to make a field read-only, mandatory, or visible
- Explain how a business rule can be used to display messages to the user
- Understand which changes are captured in Update Sets
- Create an update set to capture form changes
- Outline the process of completing and exporting an update set
- Identify the steps to retrieve, preview, and commit an update set

### Labs and Activities

- Lab 5.1: Script a UI Policy and a Business Rule
- Lab 5.2: System Update Sets
- Lab 5.3: Update Source - Update Sets

## Section 5.1: Basic Scripting

### User Story

---

As the **Service Desk Manager**,

I want to **require specific support data** for Infinity employee incidents,

So I can **speed up resolution time** for those participating in the beta testing efforts.

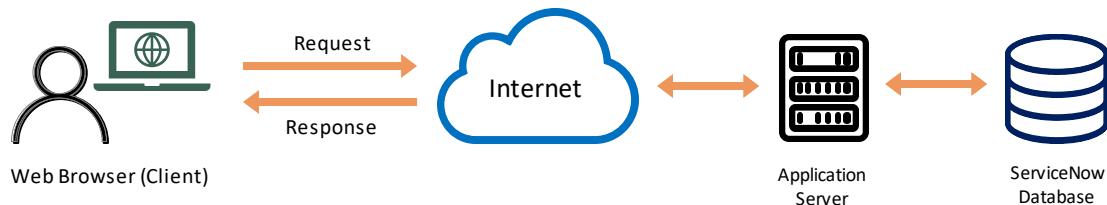


# What is Scripting in ServiceNow?

now.

Scripting in ServiceNow or Platform Scripting is the customization of an instance and/or applications using JavaScript.

JavaScript may execute on the **client side** (web browser) or the **server side** (ServiceNow database) and can fundamentally alter the baseline instance and user experience.



**Client** refers to an application or system that accesses a remote service or another computer system, known as a **server**. A **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 an Application Platform as a Service (aPaaS) model; the web browser is the client. 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).

Client to server round-trips take time and make the end-user wait for the round-trip to complete. Request + Response = Round trip.

## UI Policy and UI Policy Actions

A **User Interface (UI) Policy** is a rule that is applied to a form to dynamically change information or the form itself.

UI Policies execute on the **client** side.



Web Browser (Client)

Once a UI Policy is saved, **UI Policy Actions** determine what happens on the form, including:

- Setting a field as *mandatory* – requiring a value in order to save the record
- Setting a field as *hidden* – no longer displaying a field on the form
- Setting a field as *read-only* – preventing a user from updating its value

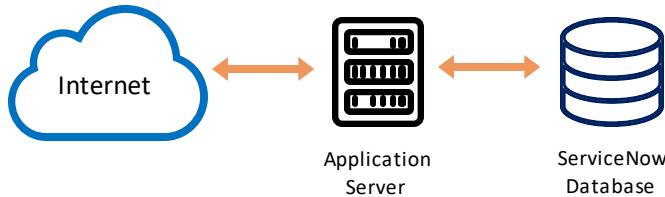
To immediately implement updates and changes to forms and lists, you can use UI Policies which allow you to add sophisticated controls without having to write scripts and define custom process flows for tasks.

Use a UI Policy to set fields on a form to:

- Mandatory or Optional
- Hidden or Visible
- Read-only or Editable

To apply a UI Policy to all *views*, set the **Global** setting to **true**.

**NOTE:** UI Policies are not about security, they are about managing the user experience.



A **Data Policy** is a rule that enforces data consistency by setting fields as mandatory and/or read-only.

Data Policy controls are similar to UI Policies but UI Policies are only enforced on data entered into a form (passing through the UI).

Data Policies are applied to *all data* entered into the platform: form (UI), Import Sets, or Web Services.

A Data Policy executes on the **server** side but can also run as a UI Policy on the client side.

Use as UI Policy on client

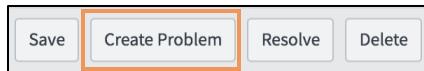
A Data Policy enforces requirements on field and record data when the data is imported into ServiceNow or when the data in an Import Set is submitted through an external system. Data Policies can be opted out for Web Services and Import Sets. A Data Policy is used to set mandatory and read-only states 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.

**NOTE:** UI and Data Policies are not about security, they are about managing the user experience.

## UI Actions

**User Interface (UI) Actions** add buttons, links, and context menu items on forms and lists, making the UI more interactive, customizable, and specific to user activities.



Adding a **Create Problem** button to an existing incident form is an example of a UI Action.

Form button	<input checked="" type="checkbox"/>
Form context menu	<input type="checkbox"/>
Form link	<input type="checkbox"/>
Form style	-- None --
List banner button	<input type="checkbox"/>
List bottom button	<input type="checkbox"/>
List context menu	<input type="checkbox"/>
List choice	<input type="checkbox"/>
List link	<input type="checkbox"/>
List style	-- None --

UI Actions can contain scripts that define custom functionality. UI Actions can be server or client side depending on the 'client' check box selection. This setting determines when a UI Action can appear.

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)

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.

**NOTE:** When the UI Actions **Active** box is checked, the UI Action is running and visible unless there is a condition met that specifies otherwise.

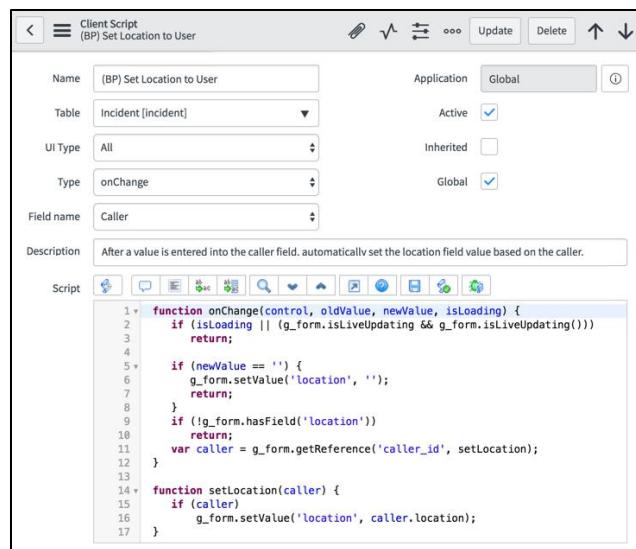
# Client Script

**Client Scripts** make “real-time” changes to the appearance of the user interface, especially forms

Client Scripts can be created to do the following:

- Automatically update the location field to reflect the value (user) entered into the caller field
- Disable the attachment link of a closed record when the form is loaded so a user is unable to add or modify attachments
- Display a notice at the top of the page to confirm a catalog request was submitted

Client Scripts execute on the **client** side



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 a user changes the priority of an incident. Client Scripts get executed on the browser, but you may also 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 through use of the list editor
- **onChange()**: runs when a particular field changes value
- **onLoad()**: runs when a form is loaded
- **onSubmit()**: runs when a form is submitted

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 the form 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.

## Script Types: Business Rule

A **Business Rule** is configured to run when a record is displayed, inserted, updated, deleted, or when a table is queried



Business Rules execute on the **server** side

Business Rules can be set to run **before** or **after** the database action has occurred

The **When** setting determines when the Business Rule executes and has the following choices:

- **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

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 are consistently applied to records regardless of how they are accessed; through forms, lists, or web services. The application of business rules through forms, lists, or web services and client scripts when fields are modified on a form is the primary difference between Business Rules and Client Scripts.

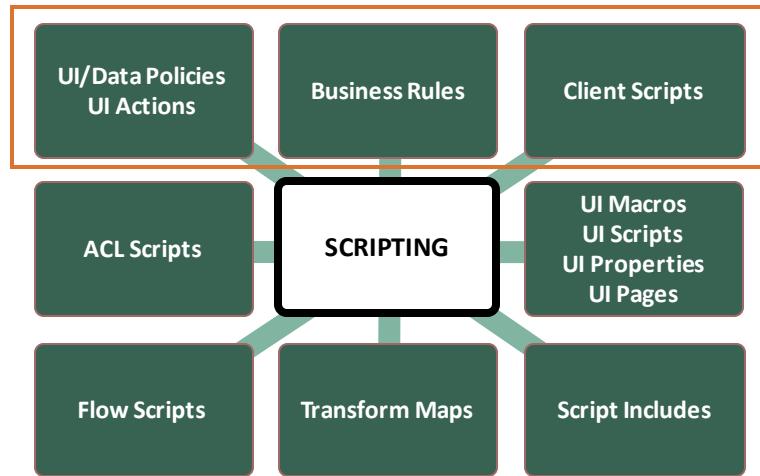
Unlike UI Policies, Business Rules are **NOT** real-time:

- They do not monitor fields on a form
- They monitor records as they are inserted or updated

The primary objective of **display** Business Rules 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.

# Scripting Areas in ServiceNow

now.



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.

More information about scripting can be found by searching [docs.servicenow.com](https://docs.servicenow.com) or [developer.servicenow.com](https://developer.servicenow.com).

# What are Plugins?

Before adding script to ServiceNow, administrators should check the list of available plugins

**Plugins** provide additional optional functionality within a ServiceNow instance

The screenshot shows the ServiceNow interface for managing plugins. On the left, there is a sidebar with filters for 'Managing' (Installed, Not Installed, Updates), 'Pricing' (Free, Paid), and 'Product family' (Other). A search bar at the top right contains the query '\*service catalog'. Below the search bar, it says '3 results for "\*service catalog"' and 'Filtered by Not Installed, Free'. The results list two items:

- Flow Designer support for the Service Catalog**: Other. It has an 'Install' button and details: ID: com.glideapp.servicecatalog.flow\_designer | Free | by ServiceNow.
- Service Catalog - ATF Tests**: Other. It has an 'Install' button and details: ID: com.glideapp.servicecatalog.atftest | Free | by ServiceNow.

A green banner at the bottom of the list area reads 'System Definition > Plugins'.

System administrators 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. Demo data can be loaded 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>.



**Time**  
**10-15m**

- Create a UI Policy with a UI Policy Action requiring the Assigned to field be populated for employee incidents
- Create a Business Rule to display an "Incident submitted" message

## Lab 5.1: Script a UI Policy and a Business Rule

As the **Service Desk Manager**,

I want to **require specific support data** for Infinity employee incidents,

So I can **speed up resolution time** for those participating in the beta testing efforts.

# Lab 5.1

⌚10-15m

## Script a UI Policy and a Business Rule

### Lab Objectives

You will achieve the following objectives:

- Create a UI Policy with a UI Policy Action
- Create a Business Rule

**Lab Dependency:** This lab uses the Employee field configured in Lab 1.3

### Scenario

Cloud Dimensions would like to continue using base functionality and available features in ServiceNow as much as possible.

One requirement for Cloud Dimensions' process improvement is to require important Infinity support data, with the goal to speed up resolution time by requiring the **Assigned to** field to be populated for all employee incidents. A confirmation for each submitted incident has also been requested.

### A. Create a UI Policy Condition

*The first step when creating a UI Policy is to identify the condition(s) under which the UI Policy should be applied and then configure the condition(s).*

1. Ensure you are logged into the instance as **System Administrator**
2. **Incident > Create New**
3. From the **Form Context Menu**, select **Configure > UI Policies**
4. Select **New**

- Fill out the UI Policy form information as shown:

**Table:** Incident [incident] (already selected)

**Short description:** Mandatory Assigned to if Employee = True

Under the **When to Apply** tab, **Conditions:**

Employee | is | true

The screenshot shows the 'When to Apply' tab of the UI Policy form. At the top, the 'Table' is set to 'Incident [incident]' and the 'Application' is 'Global'. The 'Active' checkbox is checked. Below this, the 'Short description' field contains 'Mandatory Assigned to if Employee = True' and is highlighted with a red box. Under the 'When to Apply' section, there's a 'Conditions' section with a dropdown menu. The dropdown menu is open, showing 'Employee' as the condition, followed by an 'is' operator and the value 'true', which is also highlighted with a red box. To the right of the dropdown are buttons for 'AND', 'OR', and 'X'.

- Save

## B. Create a UI Policy Action

*The UI Policy condition has been created, now the action(s) to take place when the conditions are met should be configured (i.e. Assigned to is Mandatory)*

- Scroll down to the **UI Policy Actions** section, then select **New**
- Enter the following information on the **UI Policy Action** form:

**Field name: Assigned to**

**Mandatory: True**

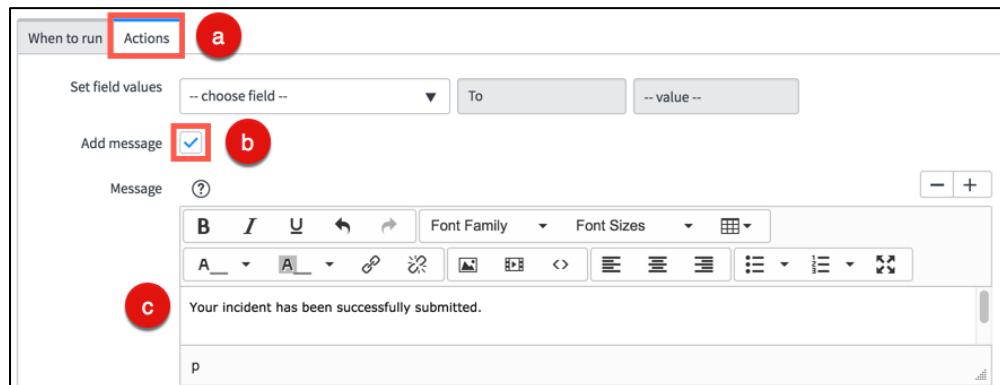
The screenshot shows the 'UI policy' section of the UI Policy Action form. The 'UI policy' dropdown is set to 'Mandatory Assigned to if Employee = True'. The 'Table' dropdown is set to 'Incident [incident]'. The 'Field name' dropdown is set to 'Assigned to' and is highlighted with a red box. To the right, there are several configuration options: 'Application' is 'Global', 'Mandatory' is set to 'True' (highlighted with a red box), 'Visible' is 'Leave alone', 'Read only' is 'Leave alone', and a 'Clear the field value' checkbox is empty.

- Select **Submit** to create the UI Policy Action.

## C. Create a Business Rule

In this section of the lab, you will create a Business Rule to display the alert, “Your incident has been successfully submitted” to all users who submit an incident, thus improving the overall user experience.

1. On an incident form, select **Configure > Business Rules** from the form context menu
2. Select **New**
3. Review the form annotations and then toggle them off using **More Options** on the form header.
4. Fill out the Business Rule form as shown:  
**Name:** Alert - Incident Submitted  
**Insert:** [check the box]
5. Add a message:
  - a) Click the **Actions** tab
  - b) Check **Add message**
  - c) Add message text: **Your incident has been successfully submitted.**



The screenshot shows the 'Actions' tab of a Business Rule configuration page. At the top, there are tabs for 'When to run' and 'Actions'. The 'Actions' tab is selected and highlighted with a red box and labeled 'a'. Below the tabs, there are two main sections: 'Set field values' and 'Add message'. Under 'Set field values', there is a dropdown menu labeled '-- choose field --' and two input fields: 'To' and 'Value'. Under 'Add message', there is a checked checkbox labeled 'b'. Below these sections is a 'Message' area with a rich text editor toolbar. The message text 'Your incident has been successfully submitted.' is entered into the text area. A red circle with the letter 'c' is placed over the bottom of the message text area.

6. Select **Submit**

## Lab Verification

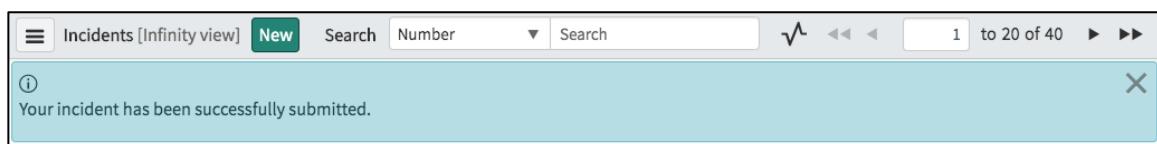
### Confirm New UI Policy is Working

1. Navigate to **Incident > Open** and open any active incident record.
2. Make sure the **Employee** field is on the form.

**Note:** If the **Employee** field is not visible, select the **Infinity** view from the form context menu.
3. Select the **Employee** field.
4. Notice that the **Assigned to** field is now mandatory.
5. Uncheck the **Employee** field and notice the Assigned to field is no longer mandatory.

### Test Your Business Rule

1. **Incident > Create New**
2. Fill out the incident form, including values for all mandatory fields.
3. Select **Submit**
4. The new Business Rule displays your message at the top of the list:



**Fantastic! You created a UI Policy and UI Policy action for employee incidents and a Business Rule to display a message for new incidents!**

## Knowledge Check

What is the language used for scripting in ServiceNow?

- **JavaScript**

Which type of scripts run on the browser?

- **Client-side scripts: UI Policies and Client Scripts**

Which script runs when a record is displayed, inserted, updated, deleted, or when a table is queried?

- **Business Rule**



Does a business rule run on the client side or the server side?

- **Server**

What attributes of a field can a UI Policy Action change on a form?

- **Mandatory**
- **Visible/Hidden**
- **Read Only**

What provides additional functionality within an instance?

- **Plugins**

## Section 5.2: Migration and Integration

### User Story

---

As a **System Administrator**,

I want to **move configuration changes** from one instance to another

So I can **easily implement updates** in our production instance of ServiceNow.

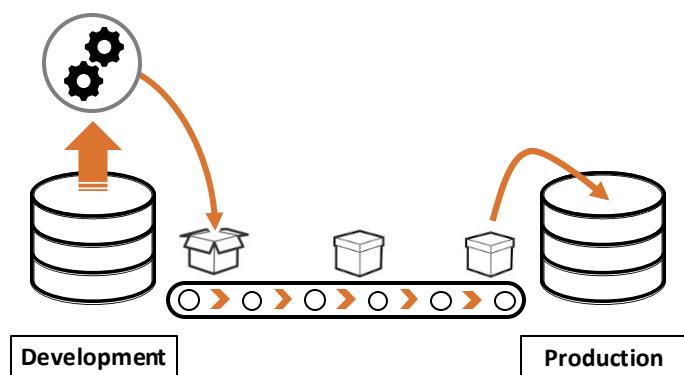


# Update Sets

An **Update Set** is a group of customizations that can be moved from one instance of ServiceNow to another.

Update Sets allow administrators to group a series of changes into a named set and then move them as a unit.

Every instance of ServiceNow has a **Default Update Set**, however, admins should use **named** update sets for moving customizations between instances.



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.

It is recommended to avoid using the Default Update Set for moving customizations between instances. Instead, use a named Update Set.

## What is Captured in an Update Set?



### Process Records

- Business Rules
- Client Scripts
- Fields
- Forms and Form Sections
- Report Definitions
- Tables
- Views
- Roles
- Published Workflows



### Data

- New Data Records
- Modified Data Records
- Tasks
- Modified Cls
- New Users and Groups
- Schedules
- Scheduled Jobs
- Homepages\*

What is captured in an Update Set is typically a customization or a configuration change and does not include changes to data records. For example, update sets track report definitions and related data such as data source, report type, style, and sharing settings. However, the actual data and report output are not tracked in an update set.

\*Homepages are **not** captured in an Update Set but can be manually added by:

- Navigating to **Homepage Admin > Pages**,
- Right-clicking on a homepage record, then
- Selecting **Unload Portal Page**.

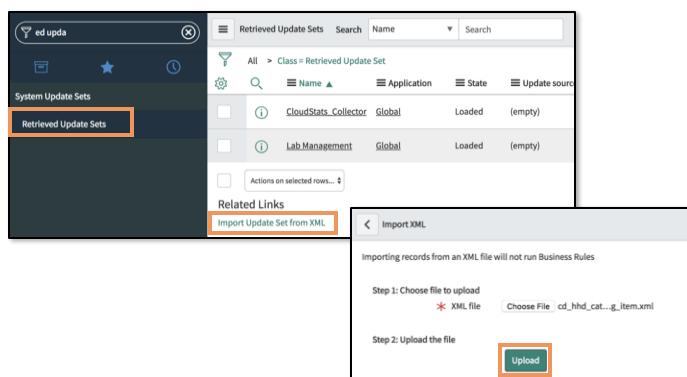
**NOTE:** Data is not captured in an Update Set. Examples: a new incident or new change record would not be in an Update Set.

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.

# Applying an Update Set

## 1. Retrieve

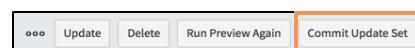
- System Update Sets > Retrieved Update Sets
- Click Import Update Set from XML
- Choose a file to upload
- Upload the file



## 2. Preview



## 3. Commit



The typical process of retrieving an Update Set includes:

- Retrieve
- Preview
- Commit

Other best practice recommendations include using the Preview to verify there are no conflicts with committing the Update Set.

**IMPORTANT:** Both instances must be on 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. You can clone the production instance onto the sub-production instance.

**NOTE:** Newest change will always overwrite older changes.

# Introduction to Integrations

now.

To share data between ServiceNow and an external system, ServiceNow integrates with many third-party applications and data sources.

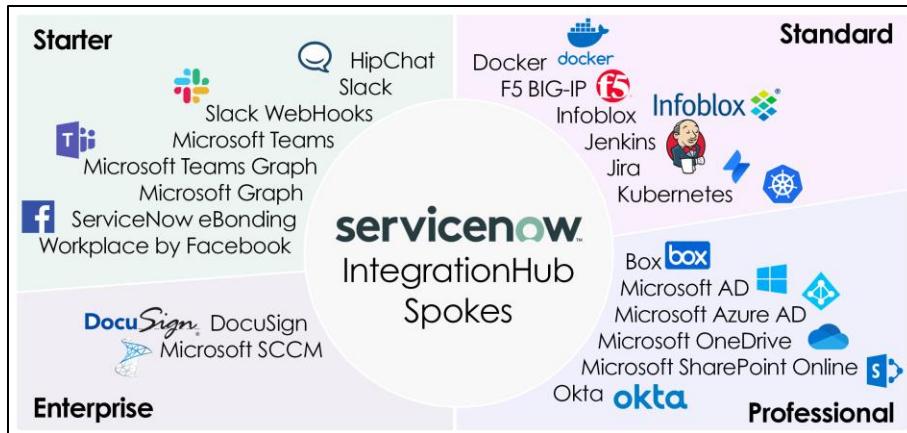
Standard integrations for ServiceNow include:

- 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.



**IntegrationHub** provides a single solution to quickly integrate with third party services to build and share content. Integrations are referred to as *spokes* and can be easily configured to integrate without scripting.

Examples:

- Post a message and details in a Slack channel when P1 incident is created
- Create users and groups in Microsoft Active Directory
- Retrieve, create, update, or delete data on a server using REST

One major benefit of IntegrationHub is that it reduces the need for code, while ensuring discoverability and reuse.

It also features scale and control mechanisms that are designed to never fail, including an extensibility framework which enables Applications to easily plug and play providers.

Use IntegrationHub to extend the Flow Designer to call 3<sup>rd</sup> party systems such as automating Microsoft Services and infrastructure using PowerShell and REST.



**Time**  
**10-15m**

- Review an Update Set
- Create a new Update Set
- Make platform changes and capture them in the new Update Set

## Lab 5.2: System Update Sets

As a **System Administrator**,

I want to **move configuration changes** from one instance to another

So I can **easily implement updates** in our production instance of ServiceNow.

# System Update Sets

Lab

5.2

10-15m

## Lab Objectives

You will achieve the following objectives:

- Review an Update Set
- Create another Update Set
- Make platform changes and capture them in the new Update Set

## Scenario

As Cloud Dimensions system administrator, review the Default Update Set to identify configuration changes that have been captured throughout class.

Then create a new Update Set and capture additional changes that will be transferred to another ServiceNow instance in the next lab.

### A. Review an Update Set

*Throughout class, a lot of work you have completed has been captured in the instance's default update set. We will review some of these updates to better understand what is captured in an update set.*

1. Logged into the instance as System Administrator, open **Settings** from the banner frame:



2. Select the **Developer** tab

3. Toggle on the switch for **Show update set picker in header**:



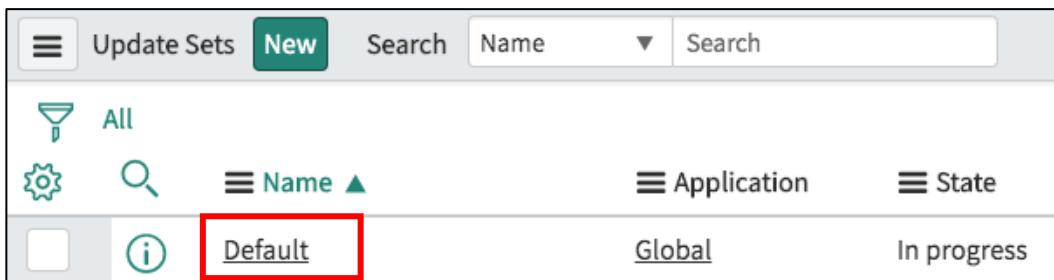
4. Close out of the System Settings window and return to your main instance screen. You will now see a drop-down menu in the banner frame, next to the user menu:



This is the **Update Set Picker** menu. It allows you to quickly select an Update Set to capture platform configuration changes.

5. **System Update Sets > Local Update Sets**

6. Locate and open the **Default** Update Set within the **Global** Application:



**Note:** Alternatively, you could click on the **View current Update Set** icon, located next to the Update Set Picker to quickly open the current Update Set's record:



7. Notice how many total **Customer Updates** have been collected.

*How many updates are there?*

*What items are captured in the Update Set that were created in class?*

*What items are not captured in the update set but were created in class?*

**HINT:** Sort the Customer Updates using the Updated by column.

## B. Create an Update Set

Now that you have seen the types of configuration changes Update Sets capture, create a new Update Set and make additional configuration changes.

1. **System Update Sets > Local Update Sets**
2. Select **New** to create a new local update set.
3. Complete the form as follows:

Name: **Catalog Item Form Changes**

Description: **Updated and Updated by fields added after Active field on Catalog Item form.**

4. Select **Submit and Make Current**
5. Close the confirmation message indicating the current update set has been changed

(i) Your current update set has been changed to Catalog Item Form Changes [Global] 

6. Confirm the Catalog Item Form Changes update set is shown in the Update Set picker:

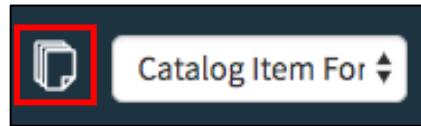


## C. Modify the Catalog Item Form

1. **Service Catalog > Catalog Definitions > Maintain Items**
2. Find and open the **Infinity** item record.
3. From the **Form Context Menu**, select **Configure > Form Layout**
4. Make the following layout changes:  
  
Add **Updated after Active**  
Add **Updated by after Updated**
5. **Save** the form configuration changes.

## D. Mark Update Set Complete

1. Click the **View current Update Set** icon, next to the update set picker, to open the update set record



2. Notice on the **Customer Updates** tab that a new record is captured:

Customer Updates (1) <a href="#">Update Set Logs</a> <a href="#">Child Update Sets</a>						
Customer Updates		Search	Created	▼	Search	
Filter	Update set = Catalog Item Form Changes	Created	Type	View	Target name	Updated by
<input type="checkbox"/>		Form Layout	Catalog Item	admin		

3. Change the Update Set State from In progress to **Complete**

A screenshot of the ServiceNow 'Update Set' edit screen. The 'Name' field is populated with 'Catalog Item Form Changes'. The 'State' dropdown menu is open, showing three options: 'In progress' (selected), 'Complete' (highlighted in blue), and 'Ignore'. The 'Parent' field is set to 'Ignore'.

4. **Save** (do not Update) the update set record.

**Note:** With the status of **Complete**, this Update Set is now ready to be exported and later retrieved from another ServiceNow instance.

5. In the *Related Links* section, select **Export to XML**.

A screenshot of the 'Related Links' section. The 'Export to XML' link is highlighted with a red box.

## Lab Verification

1. Verify you have one *Form layout* update on the *Catalog Item* table

Customer Updates (1)		Update Set Logs	Child Update Sets
		Customer Updates	Search
		Created	▼
	Update set = Catalog Item Form Changes		
	Created	Type	View
		Target name	Updated by
	[REDACTED]	Form Layout	Catalog Item
			admin

2. Confirm the exported .xml update set file is saved to your local computer. The file name should begin with: sys\_remote\_update\_set\_

For example: sys\_remote\_update\_set\_0b6eb5c77d1fb7007f44f6e0330f520a.xml

***Wow! Now you know how to create a local update set and make it the current update set, view platform changes in the update set, complete the update set, and export it to XML so it can be retrieved by another ServiceNow instance – IMPRESSIVE!***

## Knowledge Check

What is used frequently to move customizations from one instance to another?

- **Update Sets**

Which types of records are not captured in an update set?

- **Data records**

Is a homepage captured in an update set?

- **Homepages are not automatically included, but can be manually added**

What are the steps for applying an update set to an instance?

- **Retrieve**
- **Preview**
- **Commit**

Which ServiceNow product provides the ability to quickly integrate with 3<sup>rd</sup> party applications without scripting?

- **IntegrationHub**



## Section 5.3: Developer Tools

### User Story

---

As an **Application Developer**,

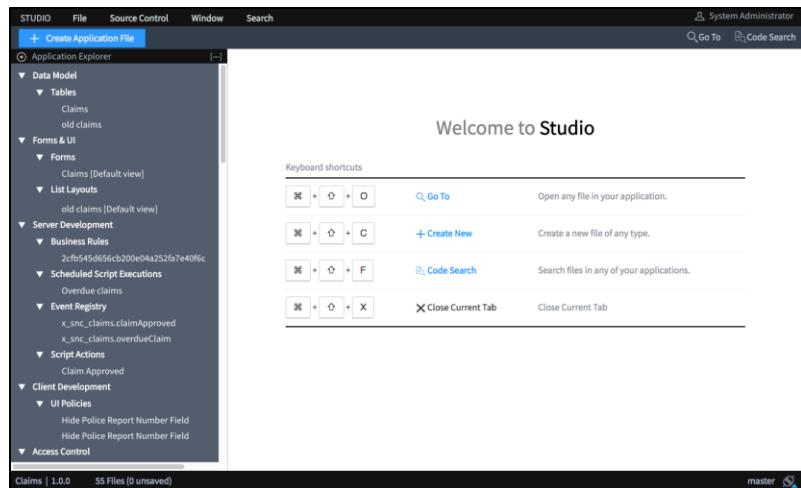
I want to **retrieve configuration changes** directly from an existing instance of ServiceNow

So I can **avoid downloading and uploading** update sets on my local machine.



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.

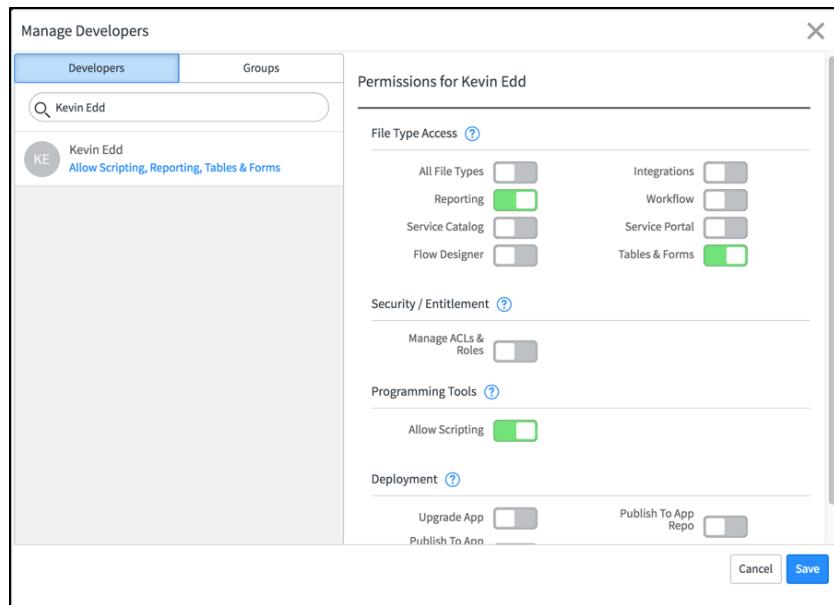
With Studio, application developers can:

- 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**

# Delegated Developers

**Delegated developers** are non-administrator users and groups which are assigned one or more permissions to develop applications

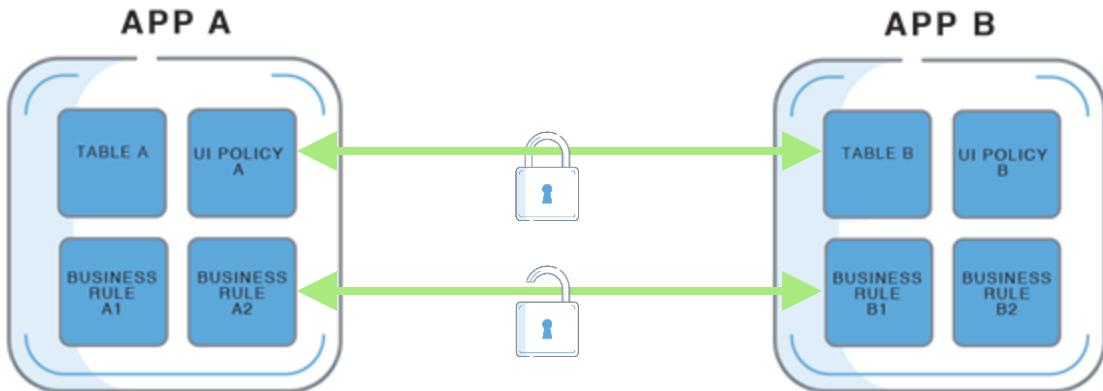
Each permission grants one or more delegated-development-specific roles to retain control over the system without having the admin role



In addition to deployment permissions, delegated developers can be granted the following permissions:

- **All File Types:** Grants the developer access to all application file types including some not granted by the other options
- **Integrations:** Grants the developer access to web service APIs, REST APIs, and data sources
- **Reporting:** Grants the developer access to reports and scheduled reports
- **Workflow:** Grants the developer access to the Workflow Editor and Activity Creator
- **Service Catalog:** Grants the developer access to catalog related file types such as catalog items, record producers, and variables
- **Service Portal:** Grants the developer access to Service Portal editors and tools
- **Flow Designer:** Grants the developer access to the Flow Designer design environment to create flows and actions. Script action steps require the **Allow Scripting** permission
- **Tables & Forms:** Grants the developer access to model and layout related file types such as table columns, form layout, and list layout
- **Manage ACLs & Roles:** Grants the developer access to security-related file types such as access controls and user roles
- **Allow Scripting:** Grants the developer write access to script fields such as those in business rules, client scripts, and Flow Designer script action steps

## Application Scopes



- Application scoping protects applications by identifying and restricting access to available artifacts 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.

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.

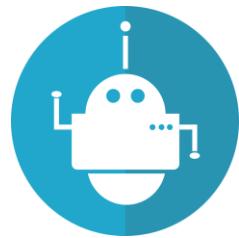
The system adds a namespace identifier to the front of application artifacts such as tables, scripts, and configuration records.

## Automated Test Framework

Consider using the Automated Test Framework (ATF) to create and run automated tests on your ServiceNow instance after modifying it

These tests provide enough flexibility to help confirm the instance still works as designed, including:

- Mimicking user actions with no scripting, such as opening a form, setting field values, etc.
- Searching for a catalog item, adding an item to a shopping cart, etc.
- Testing business rules, script includes, etc.
- Using REST requests to create, retrieve, update, or delete records



Tests include a series of steps that the test attempts to execute when ran. Once run, the Automated Test Framework creates a Test Results record, which is available through a related list on the test record.

Additionally, all test records are defined with test steps - the individual steps and the order in which the test should execute them. A test step includes an action to take and the data needed to take that action. ATF includes a number of a default set of step types (or step configs), but custom types can also be defined.

If a test includes steps that involve a form or other user-interface element, it will run the steps in a browser tab or window on-demand, or scheduled at a set time. Running tests in a browser is especially useful to watch the test execute in real-time.

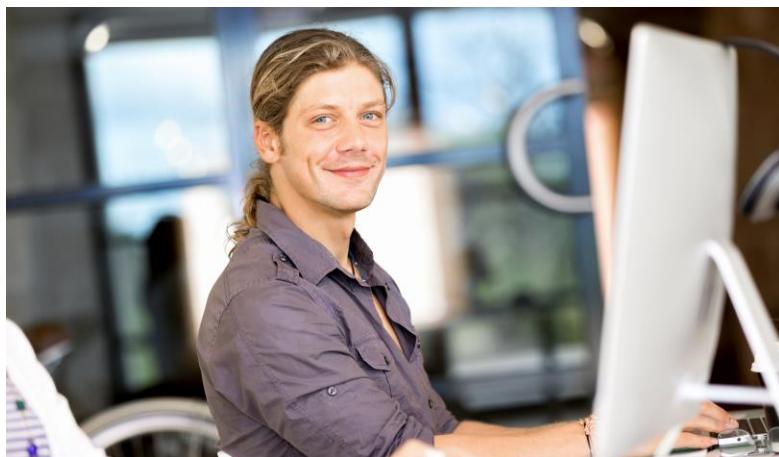
# Documentation: Developer

now

The screenshot shows the ServiceNow developer documentation homepage. At the top, there's a navigation bar with links for FEEDBACK, LOG IN, REGISTER, MORE SITES, and a search bar. Below the header, a banner says "Get your developer instance" with a "Sign Up" button and a photo of a smiling man with glasses. The main content area has three sections: "Develop Enterprise Applications" (with a sub-section about harnessing the strength of the platform), "Learn" (with a link to online training), "Build" (with a link to create and extend apps), and "Discuss" (with a link to the developer community). To the right, a detailed API reference for the `GlideRecord` class is shown, including its constructor (`GlideRecord(String tableName)`), methods like `addOrderBy`, `addQuery`, and `getNext`, and parameters for the constructor.

[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.



**Time**  
**10-15m**

- Sign up for a developer instance
- Define an Update Source
- Retrieve, preview, and commit the update set from your student instance

## Lab 5.3: Update Source - Update Sets

As an **Application Developer**,

I want to **retrieve configuration changes** directly from an existing instance of ServiceNow

So I can **avoid downloading and uploading** update sets on my local machine.

# Update Source – Update Sets

Lab

5.3

10-15m

## Lab Objectives

You will achieve the following objectives:

- Sign up for a developer instance
- Define an Update Source
- Retrieve an Update Set, Preview and Commit platform changes to the developer instance

## Scenario

To end class, you will sign up for a developer instance on the ServiceNow Developer Portal.

You will also practice the procedure of retrieving a completed update set from one instance of ServiceNow and committing the platform changes to the developer instance.

This emulates the experience of taking configuration changes made to a development or testing instance and pulling them into a production instance.

**Lab Dependency:** Requires the completion of Lab 5.2: System Update Sets

## A. Register for the ServiceNow Developer Program

*Even if you are not a developer or plan to create custom applications in ServiceNow, you can still enjoy access to your very own instance of ServiceNow! Your developer instance will be used to complete the Capstone Project (Lab 6), plus you can revisit topics covered in the class at any time!*

1. From your web browser, navigate to <https://developer.servicenow.com/>
2. Select **Register** from the upper right-hand menu, fill out the form, then read and agree to terms of use before clicking **Submit**.
3. Look for, and open, an email from ServiceNow (signon@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 2, 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 'My Learning' section of the ServiceNow interface. It displays two learning plans: 'New to ServiceNow' (0/18 Modules) and 'ServiceNow Application Developer' (0/23 Modules). Each plan has a 'Resume' button. To the right, a sidebar titled 'My Instance' contains text about requesting a free instance and a 'Request Instance' button, which is highlighted with a red box.

**Note:** If this page does not display, you can click on the ServiceNow logo (top, left) 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 the *most recent version* is recommended.

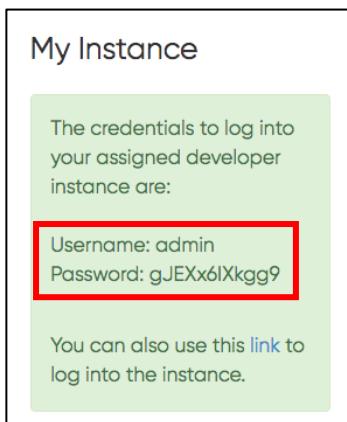
The screenshot shows a selection screen for choosing a ServiceNow instance version. It features four cards with icons representing different cities: Kingston (building with palm tree), London (Big Ben and Palace of Westminster), Madrid (Royal Palace), and New York (Empire State Building). Below each icon is the city name and a link to 'Release Notes'.

City	Icon Description	Release Notes Link
Kingston	Building with palm tree	<a href="#">Kingston Release Notes</a>
London	Big Ben and Palace of Westminster	<a href="#">London Release Notes</a>
Madrid	Royal Palace	<a href="#">Madrid Release Notes</a>
New York	Empire State Building	<a href="#">New York Release Notes</a>

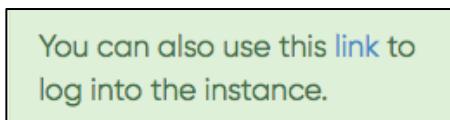
## 10. Select Request Instance

11. Capture the *Username* and *Password* shown in **My Instance**.

**IMPORTANT:** These credentials will not be displayed again once you leave this page.



12. Under the instance log in credentials, select the **link** to open the URL for your instance in another tab.



13. Using the password captured in step 11, change the temporary password to a personal password to sign in to your instance as System Administrator.

## B. Define an Update Source

1. Within your developer instance, navigate to **Service Catalog > Catalog Definitions > Maintain Items**
2. Find and open any item with a *Category* of **Hardware**

Catalog Items						
	New	Search	for text	▼	Search	
	All > Type != Bundle > Class != Order guide > Type != Package > Class != Content Item > Category = Hardware	Name	Short description	Active	Roles	Catalogs
<input type="checkbox"/>	<a href="#">Standard Laptop</a>	Standard Laptop	Lenovo - Carbon x1	true		Service Catalog
<input type="checkbox"/>	<a href="#">Developer Laptop (Mac)</a>	Developer Laptop (Mac)	Macbook Pro	true		Service Catalog

3. Observe the **Updated** and **Updated by** fields are not displayed on the form.

The screenshot shows the 'Catalog Item' form for a 'Standard Laptop'. The 'Name' field contains 'Standard Laptop'. The 'Application' field is set to 'Global'. Under 'Catalogs', 'Service Catalog' is selected. In the 'Category' section, 'Hardware' is chosen. The 'Active' checkbox is checked. Under 'Availability', 'Desktop and Mobile' is selected. A note at the bottom states 'Lenovo - Carbon x1'. The top right features standard buttons: Update, Copy, and Try It.

## C. Import Update set from XML

1. In your *developer* instance, navigate to **System Update Sets > Retrieved Update Sets**
2. Scroll to the *Related Links* section and select **Import Update Set from XML**



3. Select **Choose File**
4. Locate the file downloaded in *lab 5.2* when you exported the update set to xml
5. Select **Upload**
6. Select the *Catalog Item Form Changes* update set with the state of *Loaded*.

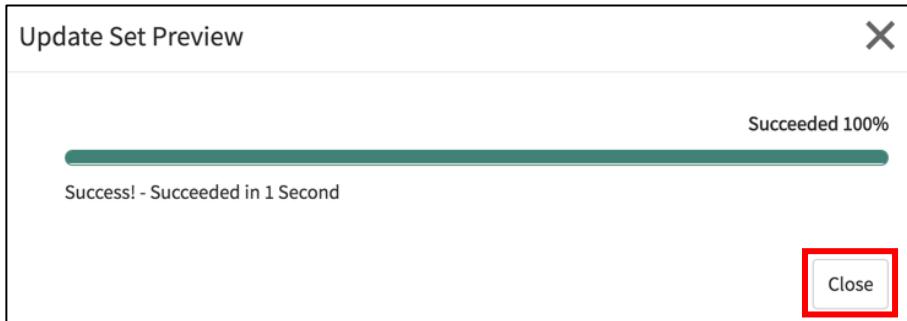
The screenshot shows the 'Retrieved Update Sets' list view. The table has columns for Name, Application, State, and Update source. One row is selected, showing 'Catalog Item Form Changes' in the Name column, 'Global' in Application, 'Loaded' in State, and '(empty)' in Update source. The entire row is highlighted with a red box.

7. Select the **Preview Update Set** button.

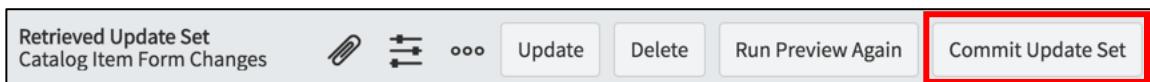


**Note:** Previewing compares an update set retrieved from a remote instance to updates on the local instance to detect potential problems.

8. Select the **Close** button from the Update Set Preview dialog box once the preview is complete.



9. Select **Commit Update Set**.



**Note:** Committing an update set applies all changes to the instance and creates a local copy of the update set that contains an update record for every change.

10. Close the **Update Set Commit** progress pop-up once it has reached 100%.

## D. Locate Retrieved Update Sets

1. **System Update Sets > Retrieved Update Sets**
2. You should see the Catalog Item Form Changes update set with a **Committed** state:

Retrieved Update Sets		Search	Name	▼	Search
All > Class = Retrieved Update Set					
		Name	Application	State	Update source
<input type="checkbox"/>		Catalog Item Form Changes	Global	Committed	(empty)

## Lab Verification

1. Navigate to Service Catalog > Catalog Definitions > Maintain Items
2. Select any item with a *Category* of **Hardware** (e.g. Standard Laptop)
3. Verify the **Updated** and **Updated by** form fields are visible:

Name	Standard Laptop	Application	Global	<a href="#">(i)</a>
Catalogs	<a href="#">Service Catalog</a>	Active	<input checked="" type="checkbox"/>	
Category	Hardware	<a href="#">(i)</a>	Updated	2016-03-04 16:53:20
			Updated by	amanjit.johal@snc

***Good job! You have learned how to bring over configuration changes from another instance by retrieving, previewing, and committing the update set!***

## Knowledge Check

Which ServiceNow tool is used by developers to create custom applications?

- Studio

A non-administrator user who is assigned one or more permissions to develop applications is called:

- **Delegated Developer**

Which site offers help and training to ServiceNow developers?

- <https://developer.servicenow.com>



What protects applications by identifying and restricting access to available files and data?

- **Application Scoping**

How can a user display the current application scope in the banner frame?

- **System Settings (gear icon)**
- **Developer tab**
- **Show application picker in header**



# Module 6

## *Capstone Project*

# Capstone Project

Show off your skills!

To help reinforce the various topics presented in **ServiceNow Fundamentals**, we present the final course component: a take-home, eight-task **Capstone Project**



## Formats

The Capstone Project is available in two formats:

1

### Challenge

- Included in your participant guide
- Provides minimum assistance to achieve end results; objectives are given but *how* you get there is mostly up to you

2

### Step-by-Step

- Downloadable from your instance
- Provides full assistance to achieve end results; detailed instructions are given to walk you through the completion of each task

The **Capstone Project: Challenge Format** is included in your participant guide. We strongly recommend that you try to solve each Capstone Project task using just the Challenge format. If you have difficulty completing a task, you can refer back to slides, notes, and labs.

If you find yourself still struggling, or prefer to complete the tasks “by the book” then no worries! Step-by-Step task solution guides are available. Locate and download these solution guides from your class lab instance. They are located under **Capstone Project** in the **ServiceNow Fundamentals Class** Knowledge Base.

# Scenario for the Capstone Project

now.

**Cloud Dimensions** has developed and tested a series of Infinity devices, used as an example throughout this course

After great success demonstrating said devices at various trade shows and conferences, Cloud Dimensions has officially begun production on a new product: the **Handheld Holographic Display**

ServiceNow will act as a Support Portal for HHD.



**Handheld  
Holographic  
Display**



Cloud Dimensions needs a software solution to support new business processes, which are still being developed, and requires that the solution be able to scale appropriately in order to support ongoing, rapid growth.

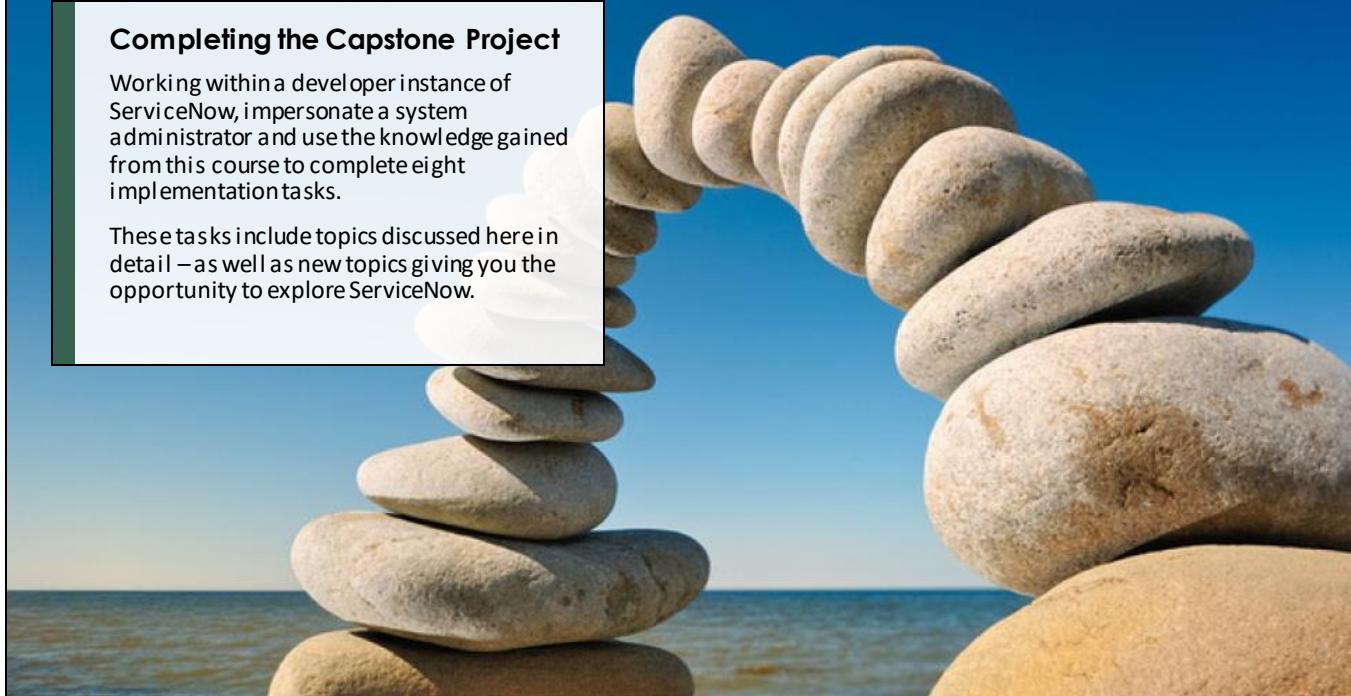
ServiceNow has proven to be the best solution to meet Cloud Dimensions current requirements and support plans for ongoing expansion. Therefore, Cloud Dimensions is ready to move forward with a ServiceNow implementation for their latest product, the **Infinity Handheld Holographic Display** (HHD).

A worldwide product launch date is aligned with the ServiceNow Go-Live date, so as a Cloud Dimensions System Administrator, you have been tasked with the responsibility to customize the Cloud Dimensions instance in support of technical needs for employees and customers alike!

## Completing the Capstone Project

Working within a developer instance of ServiceNow, impersonate a system administrator and use the knowledge gained from this course to complete eight implementation tasks.

These tasks include topics discussed here in detail – as well as new topics giving you the opportunity to explore ServiceNow.



The ServiceNow Fundamentals Capstone Project is meant to be completed on a developer instance of ServiceNow. Refer to **Lab 5.3: Update Source – Update Sets** in the course for registering in the developer program.

Remember once you start the Capstone Project: this is suppose to be a fun exercise! It is meant to test the knowledge you gained in class and, most importantly, give you a chance to look around in the ServiceNow platform.

If you start with the **Challenge** format as suggested, remember to use all of the resources discussed in class, including [docs.servicenow.com](https://docs.servicenow.com).

Enjoy!

# Capstone Project - Challenge Format

Lab  
6.1

120-150m

## Capstone Tasks Overview

The Capstone Project has been divided into eight task categories to guide your deployment:

1. Instance Customization
2. Incident Management Configuration
3. User Administration
4. Service Catalog Item Fulfillment Automation
5. Knowledge Base Management
6. Mobile UI
7. Task Assignment and Communication
8. Scheduled Reports

These tasks correlate to topics found in the ServiceNow Fundamentals materials.

To successfully complete the Capstone Project, you must select the **Capstone Project** article in the ServiceNow Fundamentals Class Knowledge Base of your lab instance. This will download a zip file to your local machine titled Capstone Project which contains relevant task files, as well as the Step-by-Step Solution guides for every task.

## Task 1: Customize Your Instance

Customize the instance to feature Cloud Dimensions branding styles that are familiar to both employees and customers. To accomplish this, you will create an organization company record to include contact information and new welcome page content.

### A. Configure Company Settings and Welcome Page

Configure the following system properties for the Now Platform® User Interface:

- **Page header caption:** *HHD Service Portal*
- **Browser tab title:** *HHD Service Portal*

- **Banner image for UI16:** *cd\_sp.jpg*
- **Header background color:** #387bcc

Create new Welcome Page Content text for all users on the login page to say:

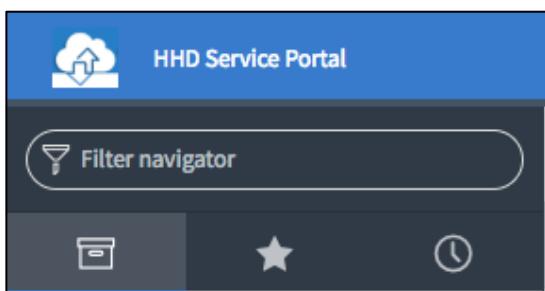
- **Short description:** *Welcome to Cloud Dimensions*
- **Text:** *Welcome to the home of Handheld Holographic Display! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.*

## B. Create New Organization Company Record

Use the **Organization** Application to create a company record with the following company information:

- **Name:** *Cloud Dimensions*
- **Phone:** *800-555-5555*
- **Street:** *3260 Jay Street*
- **City:** *Santa Clara*
- **State / Province:** *CA*
- **Zip / Postal code:** *95054*

### TASK VERIFICATION



Welcome to Cloud Dimensions  
Welcome to the home of HHD! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.

## Task 2: Update Incident Management

Modify the Incident form so that it can support a new process for troubleshooting technical issues reported by HHD users.

### A. Modify the Incident Form

1. Create a new field and add it to both the Default and Mobile View of the Incident form with the following properties:
  - **Name:** *HHD Model*
  - **Type:** *String*
  - **Field length:** *Small (40)*
2. Configure the Default View and place the **HHD Model** field beneath the **Configuration item** field
3. Configure the Mobile View and place the **HHD Model** field beneath the **Caller** field
4. Modify the **Category** field on the Incident form to include a new **HHD** choice.

### TASK VERIFICATION

The figure consists of two side-by-side screenshots of the ServiceNow Incident form interface. Both screenshots show an incident record with the identifier INC0000059.

**Left Screenshot (Default View):**

- Fields visible: Number, Caller, Category, Subcategory, Business service, Configuration item, and HHD Model.
- The "Category" field has a value of "HHD" highlighted with a red box.
- The "HHD Model" field is located at the bottom of the list of fields.

**Right Screenshot (Mobile view):**

- Fields visible: Number, Category, Priority, State, Caller, and HHD Model.
- The "Category" field has a value of "HHD" highlighted with a red box.
- The "HHD Model" field is located below the "Caller" field.

## Task 3: User Administration

Create a new user group that is responsible for troubleshooting HHD incidents and fulfilling Service Catalog HHD requests.

### A. Create Users, Groups, and Roles

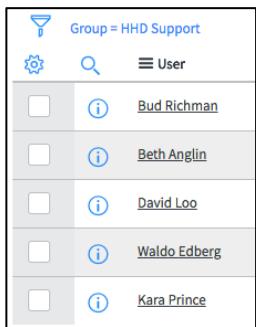
Under the existing **Service Desk** group, create a new child group called **Infinity Support** that includes the **itil** role and has Fred Luddy as the group manager.

**Create or add** the existing users to the group:

- Beth Anglin
- Bud Richman
- David Loo
- Kara Prince
- Waldo Edberg

Additionally, set Fred Luddy as the Manager under Kara Prince's user record.

### TASK VERIFICATION



## Task 4: Automate Service Catalog Item Fulfillment

Process automation can be accomplished using *Flow Designer Flows*. Create a flow to automate the fulfillment process for a service catalog item. First, you will import an HHD Prototype item into the Service Catalog to be requested, and develop a *Flow Designer* flow to support and complete the fulfillment process.

**Note:** Verify the **Flow Designer Support for the Service Catalog** plugin is **Installed** before you begin. If it is not installed, take time to install it before you attempt this task.

Configure your developer instance to accomplish the following:

## A. Import a Service Catalog Item

Import an Update Set (**cd\_hhd\_catalog\_item.xml**) containing the HHD Prototype Service Catalog item into the instance. Remember to **retrieve**, **preview**, and then **commit**!

## B. Create a New Flow Designer Flow

Develop an automated process to fulfill internal HHD Service Catalog requests. It should feature the following activities in this particular sequence:

- Approval by the requester's manager
  - The **Approved** path marks the requested item as approved with an **Approval Action**
  - The **Rejected** path marks the requested item as rejected, sends a rejection email to the requester, and then ends the workflow
- Continuing the Approved path are three Catalog Tasks:
  - **Catalog Task 1** details the steps for *ordering* the HHD item and is assigned to the **Procurement** group
  - **Catalog Task 2** details the steps for *configuring* the HHD and is assigned to the **Software** group
  - **Catalog Task 3** details the steps for *delivering* the HHD and is also assigned to the **Service Desk** group
- Upon completion of all three catalog tasks, mark the requested item complete

## TASK VERIFICATION

### A. Test the Flow – Order an HHD Prototype

---

*To test the Flow, impersonate David Loo and order the HHD Prototype. Then impersonate David's manager to approve the request.*

---

1. Impersonate **David Loo**
2. Navigate to **Self-Service > Service Catalog**
3. Select the **Hardware** category.
4. Locate and select the **HHD Prototype** item
5. Confirm the *title*, *description*, and *picture* are displayed
6. Select **Order Now**

7. Write down the **RITM** number to be used later
8. **End Impersonation**
9. Impersonate **Bud Richman** (David's manager)
10. Navigate to **Service Desk > My Approvals**
11. Open the approval request by selecting the **Requested** hyperlink
12. Select **Approve**
13. **End Impersonation**

## B. Complete the Tasks to Fulfill the Request

---

*As the System Administrator, you will test the flow logic by completing the catalog tasks rather than impersonating users in the fulfillment groups.*

---

1. Navigate to **Service Catalog > Open Records > Tasks**
2. Open the task with the short description, **Order the Infinity HHD**
3. Select **Close Task**
4. From the *List Context Menu*, select **Refresh List** (or refresh your browser)
5. Open the task with the short description, **Configure the Infinity HHD**
6. Select **Close Task**
7. Refresh the list
8. Open the task with the short description, **Deliver Infinity HHD**
9. Select **Close Task**

## C. Review Flow Execution

1. Navigate to **Flow Designer > Designer**
2. Select **Executions**
3. Open the **Infinity Workflow** execution

4. Review the status of the workflow (**Completed**) and the State of each step. Is it what you expect? If not, troubleshoot and work through the Lab Validation steps again.

Step	Action	Type	Status	Start Time	Duration
1	Catalog Item Requested		Completed	2019-08-21 20:08:33	163ms
2	Ask For Approval	Core Action	Completed	2019-08-21 20:12:22	0ms
3	Wait For Condition	Core Action	Completed	2019-08-21 20:12:22	4ms
3.1	If Requested Item Approved then	Flow Logic	Evaluated - True	2019-08-21 20:12:22	4ms
3.1.1	Create Catalog Task	Core Action	Completed	2019-08-21 20:12:22	0ms
3.1.2	Create Catalog Task	Core Action	Completed	2019-08-21 20:26:13	0ms
3.1.3	Create Catalog Task	Core Action	Completed	2019-08-21 20:28:31	4ms
3.2	Else If Requested Item Rejected then	Flow Logic	Evaluated - False		0ms
4.1	Send Email	Flow Logic	Not Run		0ms
4.2	End	Flow Logic	Not Run		0ms
5	Update Record	Core Action	Completed	2019-08-21 20:29:32	97ms

## Task 5: Update Knowledge Base

Populate the Knowledge Base with a new category that will contain two articles to support internal requesters and fulfillers. Additionally, you must abide by company security protocols and ensure the information is accessible by the appropriate parties.

**Note:** By default, all Knowledge Base articles must go through a Review Process before they are published. You will enable automatic publishing on the Knowledge Base level.

### A. Enable Automatic Publish

Adjust the settings for the IT Knowledge Base to allow for Knowledge Base articles to instantly publish upon submission – using a publish workflow to bypass any review period.

### B. Create a Requester Article

Create an article containing instructions for requesting an HHD through the ServiceNow Catalog. This article must be available to all users and located in the IT Knowledge Base under the **HHD** category. It should contain the following properties:

- **Short description:**  
*Requesting an HHD from the Service Catalog*
- **Text:**  
*To request an HHD, navigate to **Self-Service > Service Catalog**.*

*Next, click on the Hardware category and locate the HHD item. You may also use the Service Catalog search field to locate the item.*

*Click on the HHD item name to open the ordering screen where you can customize your request. Once satisfied, click the **Order Now** button on the right-hand side.*

### C. Create a Fulfiller Article

Create an article containing instructions for supporting inquiries about HHD requests through the ServiceNow Catalog. This article must be available to all users with the **itil** role and located in the IT Knowledge Base under the **HHD** category. It should contain the following properties:

- **Short description:**  
*Supporting HHD Service Catalog Requests*
- **Text:**  
*If the requester has a question about requesting an HHD, redirect them to the other Knowledge Base article: **Requesting an HHD from the Service Catalog**.*

*If the requester has placed an order and would like to know about their request, please have them contact hhdorder@cloudd.com.*

## TASK VERIFICATION

All > Workflow = Published						
	Number	Short description	Author	Category	Workflow	
<input type="checkbox"/>	<a href="#">KB0010002</a>	Supporting HHD Service Catalog Requests	System Administrator	HHD		
<input type="checkbox"/>	<a href="#">KB0010001</a>	Requesting an HHD from the Service Catalog	System Administrator	HHD		

## Task 6: Configure the Mobile UI

Configure the Mobile UI to allow users to request an HHD from the Service Catalog and create a custom application menu with a module to track and manage active HHD incidents.

### A. Publish a Service Catalog Item to the Mobile Interface

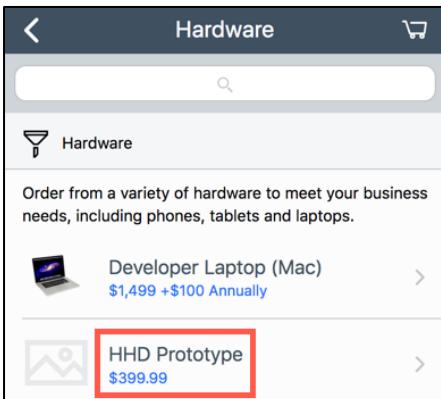
Update the HHD Service Catalog item and set its availability to display in both the Desktop and Mobile User Interfaces.

### B. Develop an Application Menu for the Mobile Interface

The application menu should be named **HHD Incidents** and limited in access to only those users with the **itil** role. It should contain one module named **Active** that displays all incident records that are active and associated with the **HHD** category.

Create a test incident by Kara Prince which has the HHD category and a short description **My HHD will not turn on.**

## TASK VERIFICATION



### Task 7: Enhance Task Assignment and Communication

Define an assignment rule that automatically assigns incidents to the Infinity Support group, if the category is HHD.

Then, develop an email notification related to new critical HHD incidents assigned to the Infinity Support group. Afterwards, test to ensure the email sends correctly.

#### A. Define an Assignment Rule

Define an assignment rule with the following details:

- **Name:** *HHD Incidents*
- **Condition:** *Category / is / HHD*
- **Assignment group:** *Infinity Support*

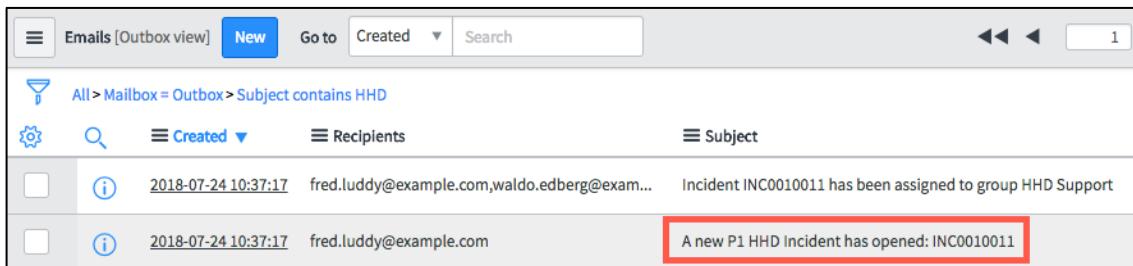
## B. Create an HHD Priority 1 Incident Notification

Create an email notification containing details about newly opened Priority 1 incidents that have **HHD** as the category. This notification should go to the current **Infinity Support** Manager only when a new Priority 1 HHD incident is created and assigned to the Infinity Support group. The notification should contain the following properties:

- **Name:** *P1 HHD Incident*
- **Subject:** *A new P1 HHD Incident has opened: \${number}*
- The **Message HTML** text should contain a collection of dynamic information listing:
  - *when the incident was opened*
  - *who opened the incident*
  - *the description of the incident*

Verify the email is sending to the Infinity Support Manager by creating a new Priority 1 HHD incident assigned to the Infinity Support group and checking the System Outbox.

### TASK VERIFICATION



## Task 8: Schedule a Report

Work with the Report Designer to create a report which displays the number of incidents that are active and tied to the HHD category. Additionally, group the data by priority.

Schedule the report by sharing it with the Infinity Support group every Monday to coincide with their incident review meeting.

## A. Create a Report

Set the following properties for the new report:

- **Name:** *Active HHD Incidents by Priority*
- **Source type:** *Table*
- **Table:** *Incident [incident]*
- **Type:** *Pie*
- **Group by:** *Priority*

## B. Schedule the Report

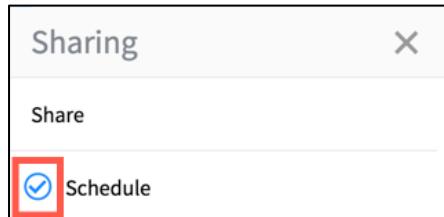
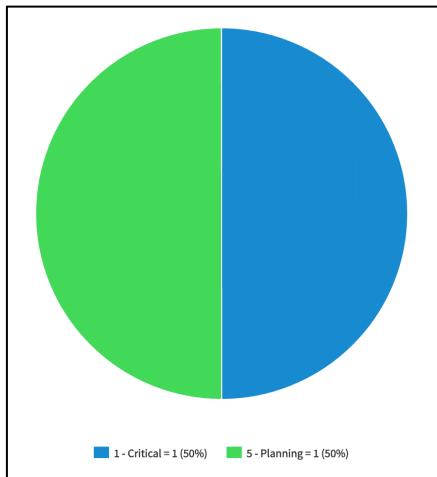
Schedule the Report to run weekly, every Monday at 8:30am, and sent to the **Infinity Support** group.

Add the following schedule details which will appear within the email containing a copy of the report:

- **Subject:**  
*Current HHD Active Incidents Count*
- **Introductory message:**  
*Please find included the current count of all active HHD incidents grouped by priority.*

*This information will be discussed during the team's incident review meeting today at 9:00am.*

## TASK VERIFICATION



[www.servicenow.com/services/training-and-certification.html](http://www.servicenow.com/services/training-and-certification.html)



2225 Lawson Ln, Santa Clara, CA 95054, USA • (858) 720-0477 • (858) 720-0479 • [www.servicenow.com](http://www.servicenow.com)

©2019 ServiceNow, Inc. All rights reserved.

ServiceNow believes information in this publication is accurate as of its publication date. This publication could include technical inaccuracies or typographical errors. The information is subject to change without notice. Changes are periodically added to the information herein; these changes will be incorporated in new additions of the publication. ServiceNow may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time. Reproduction of this publication without prior written permission is forbidden. The information in this publication is provided "as is". ServiceNow makes no representations or warranties of any kind, with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. ServiceNow is a trademark of ServiceNow, Inc. All other brands, products, service names, trademarks or registered trademarks are used to identify the products or services of their respective owners.