**Module 2 – ServiceNow Administration Fundamentals**

UNDERSTANDING DOCUMENT

**Platform Overview and Architecture**

ServiceNow is based on a Application platform as a service , It is single data model and multi-instance which means data is mingled with other companies also. It is built on advanced architecture as it has high availability architecture data centers , which provides continuous availability and it provides high backups. It provides high security for the data. It is supported by most of the browsers.

There are 3 ways to interact with platform

1.User Interface/Native UI : It is main way to interact with application which provides real time application with application navigator and many more things.

2.Mobile Interface : It has 3 different mobile apps namely Service Now agent , Now mobile , ServiceNow Onboarding. These helps user to work efficiently.

3.Portal Interface : It provides user friendly self service experience , it is advanced ui of service now which provides access to specific features, We can open it by adding sp to the url.

Supported Authentication :

1.Local database

2.SSO (Single sing on)

3.LDAP (light eight active directory account)

4.OAuth 2.0

5.Digest Token

6.Multi-factor

It supports role based access these roles assigned to user based on functions like admin, etc.. there are different components like User which is an individual that has been granted access to your ServiceNow instance , Group is a set of users who share a common purpose , Role is a collection of permissions in the Now Platform. It provides out of box data. Roles should be added to groups rather than individual user . We can impersonate user to see what features are being accessed by which role.

**User Interface and Branding**

We need to enter into service now by user name and password this opens native UI , Service now has two UI versions ui16 and UI 15 , to switch it by going to setting and switch UI versions .

There are 4 UI elements

1.Banner Frame : It the top most section which consist of the logo , which is customizable and can be used as home button, user menu which provides options like Profile , impersonate user to switch the user , elevate role for high impact actions and logout option . banner frame also contain Tools like global search , chat , help desk and system settings which has theme , accessibility , list and form , notification and developer settings.

2.Application navigator : Application navigator contains a Navigation filter which filters list of applications , All Applications option where all applications can be accessed ,and a history tab for previous searches and a favourite tab which consist of the favourites we add .

3.Content Page : The whole content which we are working on will be shown here it is more like work space.

4.Edge(available in UI 15) : Edge is only available in UI 15 , It is a like a vertical task bar located at left side of the page.

Branding can be done in service now in many ways , We have to go to UI 16 configuration where we can configure banner image , name , colour , themes and customize according to us.

**List & Filters and Forms**

List displays record from data table like incident , problem, etc.. There are different elements in list like title bar which consist list control menu which consist of view , filter , Group by , show , refresh and favourites. It has name of the table and a search bar for searching specified content. A activity stream icon which tracks activity of the table and a navigator to jump from pages , header consists a personalize list , condition builder , breadcrumbs , sort indicator , and column headings and the Data is presented in tabular format which consist of rows and columns basically these rows are the records and the columns are the attributes.

We can Import data into the list from various formats (e.g., XML) and also Export list data in formats like Excel, CSV, XML, or PDF.

Forms provide the elements you need to view change or add data to some record that stored in ServiceNow database. Every time you are viewing a record you are either viewing existing record or creating a new record. There is a form for each and every record approximately there are 4478 types of forms.

Examples : incident record form , user record form etc..

Every form consists form title , form menu , UI action buttons , header bar which Displays record type, data table, and record name. Fields which is the Main section with various data types (string, Boolean, choice, reference) Required which are mandatory to enter , Read-only in which we can only read and can not write , and Reference field for pulling values from other tables. We can change view of the form by changing view which we can access from view in menu.

We can change layout of the form using form layout can add or remove the fields in manual way and form design can create forms and arranged by using drag and drop feature

**Task Management**

A Task is any record that can be assigned or completed by a user in ServiceNow. Users create tasks and are notified as the task moves along a workflow. Tasks can be assigned to specific user or user groups.

Task Workflow goes like when a problem has raised, incident is raised and then assigned to IT team and it is routed to the one who is right person for issue. Functionalities associated with tasks are approvals, assignments, SLA(service level agreement).

The task table is a foundational table in ServiceNow, and many other task-related tables extend from it (e.g., incident, problem, request). Users need proper access to the records assigned to them. Permissions are critical for task completion. Records from extended tables are visible in the task table list. Tasks can be grouped by type (e.g., incidents, problems).

Users can manually assign tasks to groups or individuals. For example, in an incident record, you can select an assignment group and an individual user. ServiceNow provides pre-built assignment rules that automatically assign tasks based on predefined conditions. This is configured under System Policy > Rules > Assignment Rules. ServiceNow uses Predictive Intelligence which uses machine learning to predict values for fields like assignment group and assign to. This requires enabling a plugin and might require a license. It analyzes historical data to improve task assignments. Administrators and developers can create custom scripts or rules to automate task assignments based on specific business requirements.

**Notifications**

Notifications are basically the alerts which come after or before an event , there are two types outbound and inbound . Outbound Notifications are Sent by ServiceNow to users based on certain events (e.g., incident assignment). Users receive these notifications and can respond or act based on them. Inbound Actions Allows ServiceNow to process emails sent to the instance, such as creating or updating records based on email content.

System Notification Module which are accessible via Application Navigator under “System Notification.” It includes Email Notifications in which we can manage email notifications and templates. We can configure intervals, create new notifications, and manage email scripts. For mobile apps we use push notifications and Provider Notifications For agent workspace and virtual agent configurations.

We have to define the conditions under which the notification should be sent like record inserted or updated, specific event , specify the recipients, including users, groups, and fields like caller, assignment group and configure the content of the notification, including subject, body, and email templates. We can test notifications by changing relevant records like changing an incident priority to ensure that notifications are sent as expected. We also have Advanced Features like Reusable templates for email content , Watermark which Unique alphanumeric code added to each email to help identify responses and also we have Unsubscribe and Notification Preferences which allow users to opt-out or manage their notification settings.

**Knowledge Management**

Knowledge Articles are the Records in a knowledge base that provide information, such as troubleshooting tips or policy documentation.

**Creation**: Define and create knowledge articles with relevant content.

**Feedback**: Collect feedback to improve article quality.

**Import**: Import knowledge articles from external sources if needed.

Centralized Knowledge Base is the place where it Stores and categorizes articles based on business units or processes. Knowledge Articles are Referred to as KB articles, these are managed within the Knowledge Management application. Users with the appropriate roles can access, create, and maintain knowledge articles. This is managed through the "Knowledge" application in the Application Navigator. Home Page Provides a search feature for articles and displays different knowledge bases based on user access.

**Creating and Managing Articles**:

Users with the right roles can create new articles, import articles, or manage existing ones. Articles can be in various states such as unpublished, published, or retired. It also allows users without article creation access to request new articles. Users can rate articles, flag them, or provide comments to help improve content and Admins can track and manage feedback, including ratings and flagged articles.

Advanced UI Provides an enhanced, user-friendly interface for accessing knowledge articles. Customization is which Developers can customize the portal based on customer requirements.

Articles move through different stages like draft, review, and publication before becoming available to end users. Customizable workflows ensure that articles follow the appropriate review and approval process before publishing or retiring. The Knowledge Roles Includes roles for creating, editing, and managing articles like Knowledge Admin, Knowledge Manager.

For customization Workflow editor , allows the customization of workflows to fit specific organizational processes. Out-of-the-box Workflows also Includes predefined workflows for managing articles, which can be tailored as needed.

**Service Catalog**

Service Catalog is a request ordering system for services and products of any organization. It is a One stop shop to request different services provided by all the departments of Organization it is like a Categorized items help users to request the right Service and Multiple Catalogs can be created.

**Key Components of ServiceNow Service Catalog are**

**1.Order :**

**Catalog Item**:

A catalog item can be a service or product that users can request, such as software, hardware, or services like creating a new email account. For each catalog item, you can specify the price, delivery time, vendor, and other details.

**Order Guide**: A bundle of multiple catalog items grouped together for a specific request, like onboarding a new employee. It helps users order multiple items in a single request. Each item can be routed to different departments for fulfillment.

**Record Producer**:

Record producer creates records in existing tables such as incidents or change requests based on user input. It is useful for creating forms that generate records in tables, facilitating tasks like server reboots or incident reporting.

**2.Order Form**

**Variables**: Fields or questions on a form that users fill out. Examples include text fields, checkboxes, and date pickers. It is Customizable to collect necessary information from users.

**Variable Sets**: Groups of variables that can be reused across multiple forms. It is useful for common fields that appear on several forms, like "Requested for" or "On behalf of".

**3.Ordering Process:**

**Workflow**: A sequence of tasks and approvals that are processed in the background when a request is submitted. It can include custom logic and scripting.

**Flows**: A more recent addition that allows users to create automated processes with a drag-and-drop interface, reducing the need for scripting. It is Ideal for less complex automation needs.

We can view and edit catalog items, set details such as pricing and visibility, and add related articles or catalog items.Users can see and submit the form via the "Try it" button.We can configure what items are included in an order guide and under what conditions they appear. Users complete a single form that dynamically adjusts based on their selections.

**Service Catalog Record Flow**

When a user submits a request, a record is created in the sc\_request table , each catalog item within a request creates a record in the sc\_req\_item table. For each request item, catalog tasks are created in the sc\_task table for processing and approvals.

To create a catalog in ServiceNow, follow these simplified steps:

1. Create a Catalog:
   * Go to Service Catalog > Maintain Catalogs.
   * Click New, fill in details (Catalog Name, Description), and click Save.
2. Create a Category:
   * Navigate to Service Catalog > Maintain Categories.
   * Click New, enter category details, and click Save.
3. Create a Catalog Item:
   * Go to Service Catalog > Maintain Items.
   * Click New, provide item details (Name, Description), select the category, and save.
4. Add Variables to a Catalog Item:
   * In the catalog item, go to the Variables tab.
   * Click New, define the variable (e.g., Single Line Text), and save.
5. Add Variable Sets:
   * Go to Service Catalog > Variable Sets.
   * Create or use an existing variable set, add variables, and attach the set to the catalog item.
6. Design Workflow:
   * Go to Workflow > Workflow Editor.
   * Create or modify a workflow and attach it to the catalog item.
7. Use Flow Designer:
   * Navigate to Flow Designer.
   * Create or modify a flow, define the steps/actions, and attach it to the catalog item.

**Tables and Fields**

In ServiceNow, tables are fundamental components of the database, used to store data in a structured format. Each table consists of records (rows) and fields (columns). : Each row in a table represents a specific entry or record, and it contains data that is organized in fields. Columns in a table are known as fields, which hold specific types of data for each record. Field Label is the name displayed to users , Field Name is the backend name used in scripts , Field Value is the data stored in the field. Fields can be made read-only or mandatory, and their dictionary configurations can be adjusted.

**There are different types of table Relationships**

* **One-to-Many Relationship**: For example, the User table (one) has many incidents (many) associated with it.
* **Many-to-Many Relationship**: For example, the Group and Roles tables are related through a many-to-many relationship via the Group Roles table.
* **Extended Relationship**: This is where one table (child) extends another (parent). For instance, Incident and Problem tables extend from the Task table.

**There are different types of Tables**

* **Base Tables**: These are not extended from other tables. They serve as the root tables. Example: Task Table.
* **Extended Tables**: These tables inherit fields from a parent table. Example: Incident Table extends from Task Table.
* **Core Tables**: Provided by ServiceNow out-of-the-box for standard applications. Examples include Incident, Change, Problem tables.
* **Custom Tables**: Created by administrators or developers to meet specific business needs. Example: A custom table for tracking specific business metrics.

To create a table use the "New" button in the Tables module. Provide a label for the table and a name (which is auto-generated based on the label). Define the fields you need for your custom table. ServiceNow will automatically create some default fields such as sys\_id, created, and updated

**Access Control List**

ACL are the Security rules that control access to tables and records. ACLs manage which users or roles can create, read, update, or delete records. There are different types of level like table level , record level , field/ column level . There are different types of permissions are login , Application and modules , tables and records . We have different types of rules like table.none , table.\* , table.field .

When a user attempts to access a record, ServiceNow evaluates applicable ACL rules. If no matching ACL is found, access is granted. If matching ACLs are found, they are evaluated based on conditions and roles. Access is granted if conditions are met; otherwise, it is denied.

To create or edit ACLs, you need the security\_admin role. ACLs can be managed under System Security → Access Control (ACL). We can use ACLs to control access at the table, record, or field level. We can check it or test it by impersonating user.

**Data Import**

To bulk upload the data we can’t enter manually so we import data to avoid manual entry . There are different ways we can import data into servicenow are import XML in which we can only import xml data , import uses functionality of import sets and transform map we can only import Excel file with this , Import sets is core functionality to import data we can import various data into servicenow .

Import sets components are :

1.Data Source : A source of record from where the data can be imported.

2.Import Set Tables : Staging area for data before it's transformed and inserted into the target table.

3.Transform Map : Maps source data fields to target table fields.

4.Mapping Assist: Helps in visual field mapping; can be done manually or automatically.

5.Coalescing Fields: Ensures existing records are updated if matches are found; new records are inserted otherwise.

6.Target Table: Final destination where the data is inserted after processing.

**1. XML Import**

* **Exporting Data**:
  + Export the data from a ServiceNow table like Incident in XML format.
  + Edit the XML file to update specific fields like short\_description.
* **Editing XML**:
  + Open the XML file in a text editor.
  + Locate the field you want to update like example short\_description.
  + Modify the field value example, This is a demo for importing data.
  + Save the XML file.
* **Importing XML**:
  + In ServiceNow, navigate to the list view of the table ex : Problem.
  + Click on the list context menu and select Import XML.
  + Upload the edited XML file.
  + ServiceNow will automatically recognize the table and import/update the data.

**2. Import Set and Transform Map**

* + Navigate to System Import Sets in the Application Navigator.
  + Create a new import set to load data from an Excel file.
  + Click Load Data.
  + Select the file like, an Excel file you created with data.
  + Define the target table ex :Demo ACL.
  + Submit to create a staging table.
* **Creating Transform Map**:
  + Click Create Transform Map.
  + Define the transform map, selecting the target table (e.g., Demo ACL).
  + Click Mapping Assist to map fields from the staging table to the target table fields.
  + Map fields from the Excel file to the ServiceNow fields.
  + Save the transform map.
  + Click Run Transform to process the data from the staging table to the target table.

**3. Working with Import Sets**

* **Data Sources**:
  + Define different data sources (e.g., FTP servers, LDAP, CSV files).
  + Set up the connection and specify the source data.
* **Importing Data**:
  + Use the Load Data feature to bring data into ServiceNow.
  + Create a staging table to temporarily hold the data.
* **Transforming Data**:
  + Use the Create Transform Map to map and transform data.
  + Run transformations to move data from staging to the target table.
  + Check the Import Sets module for progress and errors.
  + Review the Transform History and Transform Errors for troubleshooting.
* **Advanced Options**:
  + Schedule imports for regular data updates.
  + Clean up import logs to manage system performance.

**Data Import Process:**

1. **Create Transform Map:**
   * Define how data from the import set maps to your target table.
   * Ensure the transform map is set up correctly for the fields you want to map.
2. **Transform Data:**
   * Click on "Transform" to process the import set using the transform map.
   * Check the import set log to verify the records were inserted or updated.
3. **Handle Coalesce Fields:**
   * Coalesce fields determine whether records should be updated or inserted.
   * For example, if a number field is set to coalesce, it will update existing records with that number instead of creating new ones.
4. **Update Transform Map:**
   * If you need to change how the data is handled (e.g., by using coalesce fields), update the transform map to include the necessary mappings.
5. **Test Data Import with Policies:**
   * Create and apply data policies to ensure mandatory fields are filled.
   * Verify the system’s behaviour with missing mandatory fields and adjust policies as needed.
6. **Review and Debug:**
   * If data isn't imported or updated as expected, review error messages and settings.
   * Adjust coalesce settings or data policies if needed to correct the issues.

**CMDB**

Configuration Management Database (CMDB) stores information about Configuration Items (CIs) and their relationships. examples include computers, servers, applications, and databases. We can use modules such as CI Class Manager, CMDB Groups, and CMDB Quality Builder to manage and maintain CMDB data.We can Manage CIs by editing or configuring items in CMDB.View and understand relationships between different CIs using CI Dependency View.

Make sure that the coalesce fields are correctly set to avoid unintended data duplication and also ensure that data policies are configured to prevent the import of invalid or incomplete data , before performing large-scale data imports, test with a subset to ensure that all configurations and policies work as expected.

**Configuration Item (CI) Fields are**

Name: The name of the CI.

Asset Tag: A unique identifier for the CI.

Manufacturer: The company that manufactured the CI.

Asset: Linked to the asset management table for tracking.

Class: Defines the category or type of the CI.

Company: The customer or owner of the CI.

Serial Number: The unique serial number of the CI.

Model ID: The model identifier of the CI.

Assigned To: The primary person responsible for the CI.

Comments: Any additional notes about the CI.

Fields like Hostname, OS, Domain, IP Address, etc., describe the CI’s configuration.Shows CIs related to the current CI and their connections.

CI Dashboard View provides a health overview, related CI details, and metrics like completeness, compliance, and correctness.

Base Configuration Item extends to Configuration Item. Specific tables like Hardware or Application extend further into specific classes such as Computer or Application Server.

**Usage of CMDB**

* **Incident Management**: Link incidents to CIs for effective resolution.
* **Problem Management**: Identify and resolve issues related to CIs.
* **Change Management**: Track changes to CIs.

CI Class Manager Manages CI classes in a hierarchical structure, allowing modification and extension of classes.

Manual Relationships add relationships using the Relationship Editor and Suggested Relationships use predefined relationships to maintain data integrity and relevance.

Dependency View displays a graphical view of a CI and its related CIs, showing impact and connections, useful for troubleshooting , the Dependency View provides a graphical representation of how different Configuration Items (CIs) are related.

When you open the Dependency View for a CI, it shows how various CIs are connected to the selected CI. For example, if you view a server, it might show which computers or other servers are using it.Graphical Representation displays CIs in a visual format, showing connections like "used by" or "depends on." Icons might indicate if there are incidents, problems, or maintenance tasks associated with any of the CIs.

**Integration**

**Service**Now can share data with 3rd party applications or external systems with integration some of integrations used are SSO , LDAP , Monitoring , Notifications , Events. Common integrations can be done by CMDB , incident Management , Problem Management , Change Management , User Management , Login via SSO . We have different ways of integration like web services , LDAP , EXCEL , Email .

Integration hub provides a single solution to quickly integrate with third party applications to share the data with servicenow or other system.

**Update Sets**

An update set is a group of configuration changes that can be moved from one instance to another. This feature allows administrators to group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

We have to use update sets to ensure that changes are consistent across instances like dev, test, production. We have to Perform changes in lower environments before moving them to production and avoid making direct changes in production, use update sets to mitigate risk. Items Captured in Update Sets are Form configurations, business rules, client scripts, UI policies, notifications, script includes, etc.. Items Not Captured are Task records (e.g., incidents), user data, groups, scheduled jobs, CMDB records, system properties. Ensure that instances are on the same ServiceNow version , Select the appropriate update set before making changes. Clone production instances to lower environments to maintain consistency and define a clear path for moving update sets (e.g., dev → UAT → staging → production). We have to Use clear, understandable names for update sets, review update sets for errors or warnings before committing. Capture the changes in the development instance. For testing move to UAT, test, then to staging for further validation and finally, move validated update sets to the production instance. sys\_update\_set is the place which Stores update set data, Customer Update Table Stores individual changes within an update set.

**Events**

Events are special log records the system generates when something notable has happened or certain conditions occur like user login or approve or submit. Events can be generated with Business rules , API , Flow , Workflow . We can configure action that event has occurred . we can Send notifications based on events and define actions in business rules and flows. We can execute the scripts as a result of events. If we have generate event first we have to capture it in even registry , simply go to event registry and create event specify event we can use it in workflow or in API . We can check what kind of event generated we can check in event logs it capture all events . Once event is triggered we can perform actions like sending notifications , etc..

**Platform Stats**

Stats module provides statistics for system activities that affect performance such as the execution of queries, scripts, and transactions. These stats can show information like version , cluster , build name , etc.. . It is also used for trouble shooting the problems .