ServiceNow System Administration Overview

Platform Overview

- ServiceNow Platform: Cloud-based automation for managing business processes.
- Architecture: Utilizes a multi-instance architecture with robust data security features.

Backup and Security

- Regular Backups: Supports frequent backups and recovery options.
- Security Features: Includes SAML 2.0, role-based access control (RBAC), and multi-factor authentication (MFA).

Platform Interface

- Native UI: Primary interface with real-time updates.
- Mobile Apps: Available for on-the-go access.
- Service Portal: Self-service access for users.

Supported Authentication Methods

- Methods: Local database, Single Sign-On (SSO), LDAP, OAuth 2.0, Digest Token, and MFA.

Role-Based Access Control (RBAC)

- Roles: Define permissions and access.
- Users and Groups: Manage access by assigning roles to groups for simplified management.

Managing Users, Groups, and Roles

- Out-of-the-box Tables: Use standard tables to manage users and roles.
- Role Assignment: Assign roles to groups to streamline management.
- Impersonation: Administrators can impersonate users to test access levels.

User Interface Versions and Elements

UI 15 vs. UI 16

- UI Differences: Explore the key differences between UI 15 and UI 16 elements.
- UI Elements:

  Banner Frame
  Application Navigator
  Content Frame
  Edge (UI 15 only)
  Branding and Customization (UI 16)

- Configuration: Modify settings like banner image, system name, and background color.
- Welcome Page: Customize login/logout messages for users.

Lists and Filters

List Components

- Key Elements: Includes title bar, filters, breadcrumbs, and columns.
- Filtering Data: Apply and save filters for customized data views.

Views and Grouping

- Custom Views: Create views like Default View, Major Incidents View.
- Data Grouping: Group by fields such as "State" or "Priority".

List Configuration

- Context Menu: Offers options for sorting, grouping, and visualizing data.
- Layout Modification: Customize field visibility and arrangement.

Import/Export

- Supported Formats: Export lists in formats like Excel, CSV, or PDF.
- Bulk Updates: Perform updates on multiple records simultaneously.

Incident Management Application

Forms and Fields

- Field Types: Mandatory, read-only, choice, reference.
- Saving Records: Use "Submit" or "Save" for record actions.
- Post-Save Actions: Use options like "Update" or "Create Security Incident."

Form Configuration

- Form Layout vs. Design: Basic layout modifications vs. advanced drag-and-drop customization.
- Field Management: Add and adjust new fields for forms.

Form and List Layouts

- Related Lists: Customize and configure related lists at the bottom of forms.
- Personalization: Customize form views based on specific requirements.

Templates and Scheduling

Using Templates

- Purpose: Streamlines data entry with pre-filled fields.
- Consistency: Ensure uniformity across records.

Scheduling Templates

- Automation: Automatically create records at predefined times.
- Use Case: Recurring tasks or routine updates.

Task Management

Understanding Tasks

- Definition: Tasks represent records in ServiceNow, like incidents or requests.
- Task Examples:
  - IT team resolving an issue.
  - HR team onboarding a new employee.
  - Task Workflow
- Scenario: Track the process from task creation to resolution with notifications sent to stakeholders.

Task Table and Extensions

- Core Table: The task table is extended by specialized tables like incidents and problems.
- Field Inheritance: Extended tables inherit fields from the task table.

Functionalities with Tasks

- Approvals: Can be manual or automatic.
- Assignments: Assign tasks to users or groups, manually or automatically.
- Service Level Agreements (SLAs): Track task completion timelines.

Task Assignment and Access

Manual Task Assignment

- Process: Manually assign tasks via the task form.

Assignment Rules

- Pre-built Rules: Automate task assignment based on predefined conditions.
- Custom Rules: Create tailored assignment rules using scripts.

Predictive Intelligence

- Machine Learning: Predicts task fields such as category or priority.

Service Desk Application

Purpose

- Functionality: Manage incidents, requests, changes, and problems from one module.

Modules

- Callers: Access user information.
- Incidents: Manage and track incidents.
- Knowledge: Access knowledge base records.
- My Work: View personal tasks and approvals.

Effective Task Management

Work Notes and Comments

- Updates: Track task progress and communicate via activity streams.

Notification Management

- Outbound Notifications: Notifications for record updates (e.g., assignments).
- Inbound Actions: Process responses via email.

Notification Application

Email Notifications

- Digest Intervals: Reduce email frequency by summarizing notifications.
- Digest Content: Customize content separately from regular notifications.

Modules

- Email: Manage email-based notifications.
- Push: Handle mobile notifications.
- Provider: Configure notifications for Agent Workspace and Virtual Agent.

Notification Management Overview

Notification Overview

- Out-of-the-Box Notifications: Pre-configured notifications for common events in ServiceNow, reducing setup time.
- Notification Creation: Users can create custom notifications using the "New" button to meet specific business needs.

Notification Components

- Notification Email Scripts: Predefined scripts used to customize notification content.
- Notification Categories: Organize notifications into categories for easy management; admins can create new categories as needed.
- Email Templates: Reusable email templates managed by administrators for consistency in notifications.
- Notification Filters: Filters to send notifications based on specific conditions and allow users to manage their own subscriptions.
- Email Access Restriction: Controls access to specific notifications, ensuring data security.

Advanced Modules

- Push Notifications: Allows configuration of notifications for mobile devices.
- Provider Configuration: Settings for notifications in advanced environments like Agent Workspace and Virtual Agent.

Notification Table

- Table Name: Notifications are stored in the sys_event_email_action table, where all related data can be tracked.

Notification Form Fields

- Name: The title of the notification.
- Table: The table triggering the notification, such as Incident or Problem.
- Category: Defines the classification of the notification.
- When to Send: Triggers for sending notifications:

  Record Inserted/Updated
  Event Triggered
  Flow Designer Triggered
- Who Will Receive: Notification recipients, such as:

  Users

Groups
Subscribable User/Group fields for user subscription.

- What It Will Contain: Notification content includes:
  Content Type (HTML, Plain Text)
  Attachments, Importance (High/Low), Custom Sender (From Field), Reply Address (Reply To)
  Push Messages for mobile notifications.
  Email Templates for consistent content.

Creating a Notification

Steps:

- Click "New," fill in the notification details, configure triggers (When to Send), recipients (Who Will Receive), and content (What It Will Contain).
- Example: A notification for incidents when the priority changes to "Critical," with specified recipients.

Assign and Save Notification

- Save and assign the notification, ensuring content fields like the body and subject are configured with dynamic placeholders such as incident numbers.

Testing the Notification

- Test by updating an incident's priority, and review email logs to verify the notification was delivered and received correctly.

Email Content and Features

- Unsubscribe Options: Include unsubscribe options to allow users to manage their notification preferences.
- Watermark: Each email contains a unique identifier code for replies.

Customizing Email Content

- Use Email Scripts for dynamic content with the syntax $mail_script{name}.

Inbound Email Actions

- Configuring Inbound Actions: Define actions like creating or updating records based on incoming emails. Specify the target table, action type, and necessary conditions.
- Inbound Action Fields: Define fields like Name, Target Table, Action Type, and Conditions.

Knowledge Management Overview

Knowledge Article Overview

- Definition: Provides policies, troubleshooting steps, and solutions in the form of articles.
- Benefits: Improves accessibility to information and enhances support efficiency.

Introduction to Knowledge Management

- Centralized solution for self-service, enabling the creation, categorization, and viewing of knowledge articles.

Knowledge Application

- Access: Users with the appropriate roles can manage articles, submissions, and feedback via the knowledge home page.

Knowledge Base and Article Management

- Organized: Articles are managed by departments or business units.
- Stages: Articles pass through stages: Unpublished, Retired, Published, and All.

Submissions and Feedback

- Submissions: Users can request the creation of articles if unable to create them directly.
- Feedback: Ratings, flags, and comments help improve article quality.

Feedback Management

- Tracking: Feedback mechanisms track comments and ratings for quality assurance.

Administration and Configuration

- Modules: Admins can manage knowledge base settings through modules like Guided Setup, Ratings, Search Logs, and User Criteria.

User Criteria

- Visibility Control: Defines which users can see specific articles based on roles, locations, and departments.

Diagnostics

- User Criteria Diagnostics: A tool to troubleshoot access issues with knowledge articles based on user roles.

Knowledge Base Organization

- Effective Article Management: Knowledge bases help manage articles by categorizing them based on departments or units.

Article Lifecycle

- Stages: Articles progress through stages (e.g., Draft, Published, Retired), affecting visibility and accessibility.

Configuration and Access Control

- Admins: Administrators control knowledge management settings and user access via roles and criteria.

Diagnosing User Access to Articles

- Purpose: Allows administrators to troubleshoot user access to articles by selecting users and articles to check permissions.

Knowledge Portal Overview

- Advanced Knowledge Portal: An enhanced portal with user-friendly layouts for searching and viewing articles.

Creating and Viewing Knowledge Articles

- Fields: When creating an article, key fields include the number, knowledge base, category, valid dates, and type.

Roles Associated with Knowledge Management

- Knowledge Role: Basic permissions for knowledge article management.
- Knowledge Admin: Full permissions to manage knowledge bases and articles.
- Knowledge Manager: Responsible for overseeing workflows and article management.

Knowledge Management Workflow

- Stages: Draft, approval, publication, and retirement are key stages in article management.
- Custom Workflows: Organizations can define custom workflows to suit specific needs.

Importing Articles

- Process: Articles can be imported from Word documents into the knowledge base.

Service Catalog Overview

- Function: A system for requesting services and products from different departments, similar to an online shopping platform.
- Benefits: Centralized platform for requesting, organized services, and order tracking.

Key Benefits of Service Catalog

- Unified Platform: Central location for service requests.
- Categorization: Services are logically grouped for easy navigation.
- Tracking: Users can track requests and manage their orders.

Service Catalog Application

- Roles: Manage services and products based on user roles.
- Configuration: Ability to configure catalog items and categories.
- Access: Available through the application navigator.

Catalogs

- Purpose: Manage various catalogs for different service types.
- Types:
  Service Catalog: Default catalog for end-users.
  Technical Catalog: Designed for technical services.

Modules

- Catalogs: Manage and create catalogs.
- Open Records: View requests created via the catalog.
- Maintain Categories: Manage catalog categories, including dynamic categories.
- Service Catalog Configuration: Configure services, categories, and products.

Categories

- Purpose: Organize services and products for easy discovery.
- Management: Create, edit, and manage categories with relationships (e.g., parent-child).

Catalog Items

- Purpose: Define products or services for request.
- Components:
  Variables: Fields for user input during request submission.
  Variable Sets: Reusable collections of variables across multiple catalog items.

Ordering Process

- Workflow: Admins can define complex approval and assignment workflows.

- Flows: Drag-and-drop designer to simplify process automation with minimal scripting.

Service Catalog Management

- Item Configuration: Includes catalog names, categories, workflows, and appearance customization.
- Execution Plans: Define how requests are processed, including workflows or flows.

User Interaction

- End User View: Users fill out forms, submit requests, and provide necessary input.

Order Guides

- Overview: Group multiple catalog items into a single request to streamline the ordering process.
- Functionality: Helps request multiple related services simultaneously (e.g., IT services and email setup).

Record Producers

- Function: Creates records in ServiceNow tables based on user inputs from catalog forms.
- Usage: Allows users to submit incidents or requests directly through the catalog.

Variables and Variable Sets

- Variables: Data input fields in catalog items; various types available (e.g., text, checkboxes).
- Variable Sets: Predefined sets of variables for consistent use across multiple items.

Workflow and Flow Design

- Workflow: Manage request workflows using the Workflow Editor.
- Flow Designer: Drag-and-drop interface for automating processes.

Catalog Item and Request Flow

- Request Records: Requests are stored as records (e.g., sc_request, sc_request_item).
- Process: Requests flow through stages, including approval and fulfillment.

Request Stages

- Custom Stages: Configurable request stages to suit business needs (e.g., approval, delivery).

Creating Catalogs and Items

- Catalog Creation: Define new catalogs and categories as required.

- Catalog Item Setup: Configure variables, categories, and workflows for catalog items.
- Practical Example
- Catalog Item Creation: Create catalog items with custom variables and variable sets for user input.

ServiceNow Data Structure

- Tables and Fields: Data is stored in tables, with each record containing multiple fields.
- Core Tables: Built-in tables for various purposes (e.g., Incident, Task).
- Custom Tables: Can be created for specific needs, inheriting fields from core tables.

Table Relationships

- One-to-Many: Example: A user can have multiple incidents.
- Many-to-Many: Example: Groups and roles in a many-to-many relationship.

Access Control in ServiceNow

- Custom Fields: Fields can be added to tables as needed.
- Schema Map: View relationships between tables and fields.
- Access Control List (ACL): Defines permissions for accessing and modifying records.

Data Import into ServiceNow

- Import XML: Transfer data between ServiceNow instances using XML.
- Import Sets: Import and map data from external sources into ServiceNow tables.
- Import Set Components
- Data Source: Defines where the data comes from (e.g., LDAP, files).
- Import Set Table: Temporary storage for imported data before transformation.
- Transform Map: Maps source data fields to ServiceNow table fields.
- Transform Map Creation and Field Mapping
- Process: Import data, create a transform map, and configure field mappings.
- Coalesce Fields: Identify records for update rather than creating duplicates.

Data Import and Transformation

- Coalesce Fields: Use to update existing records without duplication.
- Handling Data Policy Exceptions: Mandatory fields must be filled for data import.

Configuration Management Database (CMDB)

Overview of CMDB

- Purpose: Stores information about configuration items (CIs) and their relationships, facilitating IT asset management and dependency tracking.

Configuration Items (CIs)

- Definition: Organizational assets such as computers, servers, databases, and applications.

CMDB Application in ServiceNow

- Modules: Includes out-of-the-box tools like CI Class Manager, CI Form, CI Record, and CI Dashboard View.

CMDB Modules

- CI Class Manager: Manages CI classes in a hierarchical structure.
- CI Form: Displays detailed CI information (e.g., hardware specs, related items).
- CI Record: Contains fields such as CI name, asset tag, and configuration details.
- CI Dashboard View: Shows CI health metrics, related tasks, and stale data.

Key CMDB Tables

- Main Tables: Includes the Base Configuration Item (parent table), Configuration Item (extended table), and CI Relationship (stores CI relationships).

CMDB Table Hierarchy

- Structure: Starts with Base Configuration Item and extends into specific classes (hardware, applications) with further categories.

Usage of CMDB

- Incident Management: Involves using CIs in incidents for accurate issue resolution. CI selection is often mandatory.

CI Dependency View

- Graphical View: Visualizes the relationships between CIs, aiding in impact analysis for troubleshooting.

Practical CMDB Usage Example

- Incident Creation: Use CIs during incident creation for better tracking and routing.

Configuration Item (CI) Relationships

Viewing CI Relationships

- Viewing Relationships: Access records related to a CI (e.g., tasks, changes).

Adding Relationships

- Relationship Editor: Add and save new CI relationships.
- Automatic Relationship Creation: Update the CI relationship table for direct relationship additions.
- Suggested Relationships: Prevent misuse by suggesting appropriate relationships.
- Dependency View
- Visualizing Dependencies: Helps identify potential impacts during issues by showing CI interconnections.

CMDB Functionality

- Impact Analysis: Enables proactive issue resolution by visualizing CI relationships and dependencies.

Update Sets in ServiceNow

Overview of Update Sets

- Definition: A group of configuration changes bundled for deployment between instances.

Moving Changes with Update Sets

- Example: Capture UI action changes in an update set and replicate them across instances.

Update Sets Application

- System Update Sets: Manage, retrieve, merge, and commit update sets across instances.
- Local Update Sets: Specific to the instance currently in use.
- Merge Update Sets: Combine multiple update sets into a single one for ease of use.
- Update Sets to Commit: Commit update sets for installation within the instance.

When to Use Update Sets

- Use Cases: Ideal for replicating changes, testing configurations, and managing impactful system updates.

Captured Items in Update Sets

- Automatic Capture: Forms, business rules, client scripts, and more.
- Items Not Captured: Task records, user data, groups, scheduled jobs, and CMDB records are excluded.

Update Set Planning Process

- Version Consistency: Ensure all instances are on the same version.
- Update Set Selection: Select the appropriate update set during development.
- Instance Cloning: Maintain consistency across production and lower instances.
- Update Set Path: Define a clear promotion path for update sets.
- Commit Timing: Plan commits to minimize user impact.
- Naming Convention: Use clear, descriptive names for update sets.
- Preview and Commit: Review changes before committing to avoid issues.

Update Set Promotion Process

- Instance Flow: Move update sets through development, UAT, staging, and production environments.

System Update Set Tables

- Update Set Table: Stores update set records in XML format.
- Customer Update: Tracks specific customizations within the update set.

Events in ServiceNow

Event Generation

- Triggers: Events are generated by user actions like logins, approvals, or request submissions.

Event Processing

- Actions: Events can trigger specific actions or scripts based on conditions.

Scripting Events

- Custom Events: Use gs.eventQueue() for generating custom events.

Event Registry

- Definition: Manages system event names and actions.
- Naming Convention: Use standardized names for event clarity.

Event Log

- Functionality: Monitors system events for troubleshooting and filtering by date and time.

Platform Stats

- Functionality: Provides statistical data on system performance, such as version details, memory usage, and instance statistics.