

Orchestration

Orchestration automates simple or complex multi-system tasks on remote services, servers, applications, and hardware.

An Orchestration process can cross all management disciplines and interact with hosted services and all types of infrastructure elements. These capabilities provide a powerful system for managing IT and Business processes quickly and reliably, with best practices every time.

Orchestrated solutions aide collaboration among teams by providing reusable data and versioning for both the workflows and the activities within them. This allows subject matter experts to create activities that are consumed by numerous workflow developers. When an activity requires a change, developers can see the downstream implication of the change immediately by knowing which workflows use the activity. Well designed orchestrations never require human intervention when automation errors arise.

Orchestration provides the ability to make calls outside of a ServiceNow instance, directly to SOAP and REST web services or to systems within an enterprise's corporate firewall through the MID Server. Orchestration extends the **workflow editor** by providing these features:

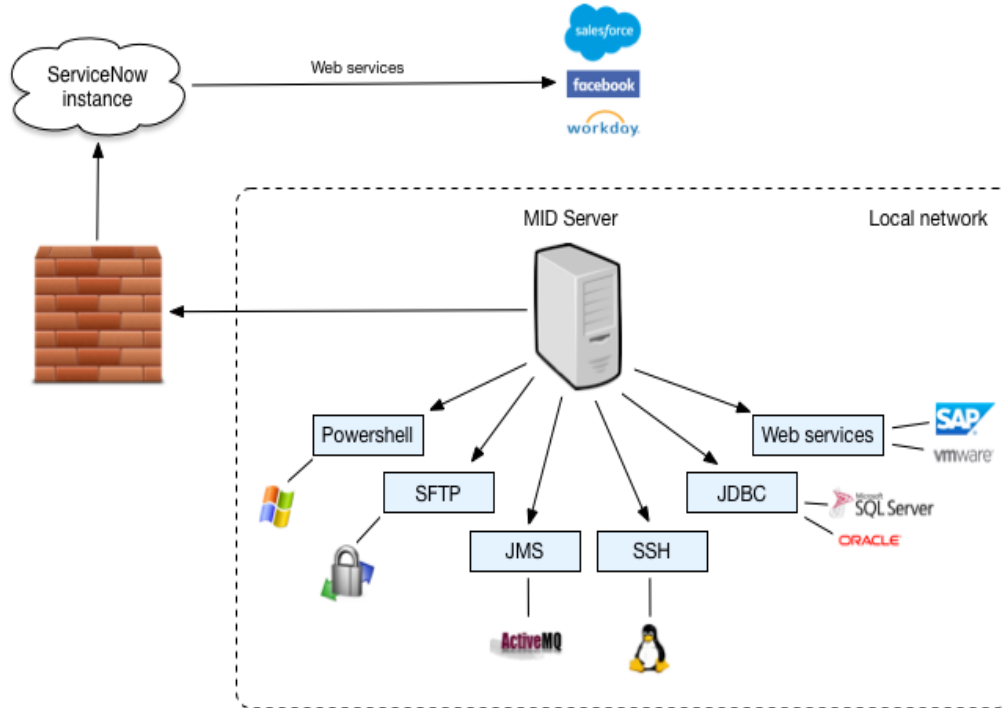
- **Activity packs** containing ready-to-use activities.
- An **activity designer** that allows developers to create custom activities without having to rely on Scripting..
- Ability to create activity packs using **scoped applications**.
- A **databus** for reusable data.

System that can be automated

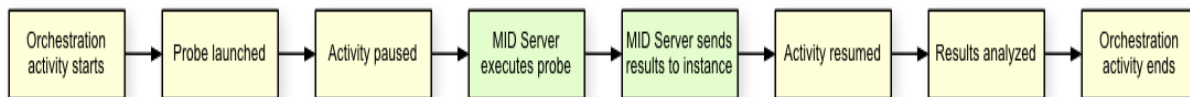
Orchestration can automate tasks such as employee onboarding, user access rights, server management, and managed file transfers. For example, you might use the Active Directory and Exchange activities provided in the base Orchestration system to set up network accounts and mailboxes for new employees.

If Orchestration does not provide the activities you need for an integration, create the necessary activities using the templates in the **Orchestration activity designer**.





When an Orchestration activity starts within a workflow, Orchestration launches a **probe** and writes a probe record to the ECC Queue. The workflow pauses as the MID Server picks up the request and executes the probe. When the probe reports back, the workflow resumes as the results are analyzed. The workflow can exit or continue at this point.



Activate the Orchestration plugin

To purchase a subscription, contact your ServiceNow account manager. The account manager can arrange to have the plugin activated on your organization's production and sub-production instances, generally within a few days.

Target Instance Select the instance on which to activate the plugin.

Plugin Name Enter the name of the plugin to activate.

- Navigate to HI.
- Click **Service Catalog**.
- Click **Request Plugin Activation**, and fill out the form.
- Click **Submit**.

Orchestration example of employee on-boarding/off-boarding

A set of six Orchestration Active Directory activities enables organizations to automate their on-boarding/off-boarding processes with auditable, self-documenting workflows that save time and eliminate mistakes.

The activities in the **Active Directory activity pack** are designed to manage user accounts and reset user passwords. The following activities cover the most common use cases administrators encounter when managing Active Directory user accounts:

- Create AD User Account
- Update AD User Account



- Remove AD User Account
- Disable AD User Account
- Query AD
- Reset AD User Password.

These activities share a common design, have complementary functionality, and share a common set of parameters. They can be used singly or together to create consistent workflows for provisioning and de-provisioning user accounts.

Updating Active Directory

An organization plans to make their ServiceNow instance the single system of record for user account data and wants to update Active Directory with the latest changes.

About this task

The solution is to create an Orchestration workflow that pushes changes from the ServiceNow user record down to the Active Directory to create a new user record or update an existing record. This is accomplished by creating an Orchestration workflow that can create and update records in Active Directory based on the data in the ServiceNow User [sys_user] table.

Note: This example workflow assumes that ServiceNow is configured for LDAP and an LDAP server is configured to accept the new user accounts. The Active Directory user management activities are not dependent on LDAP, but the presence of LDAP makes this example workflow much easier. You must provide the domain controller's IP address to the workflow, either by hardcoding it, adding another workflow input, or using a script to look it up from the CMDB.

Procedure

Navigate to **Workflow > Workflow Editor**.

In the **Workflow** tab, click the + icon to create a new workflow using these variables:

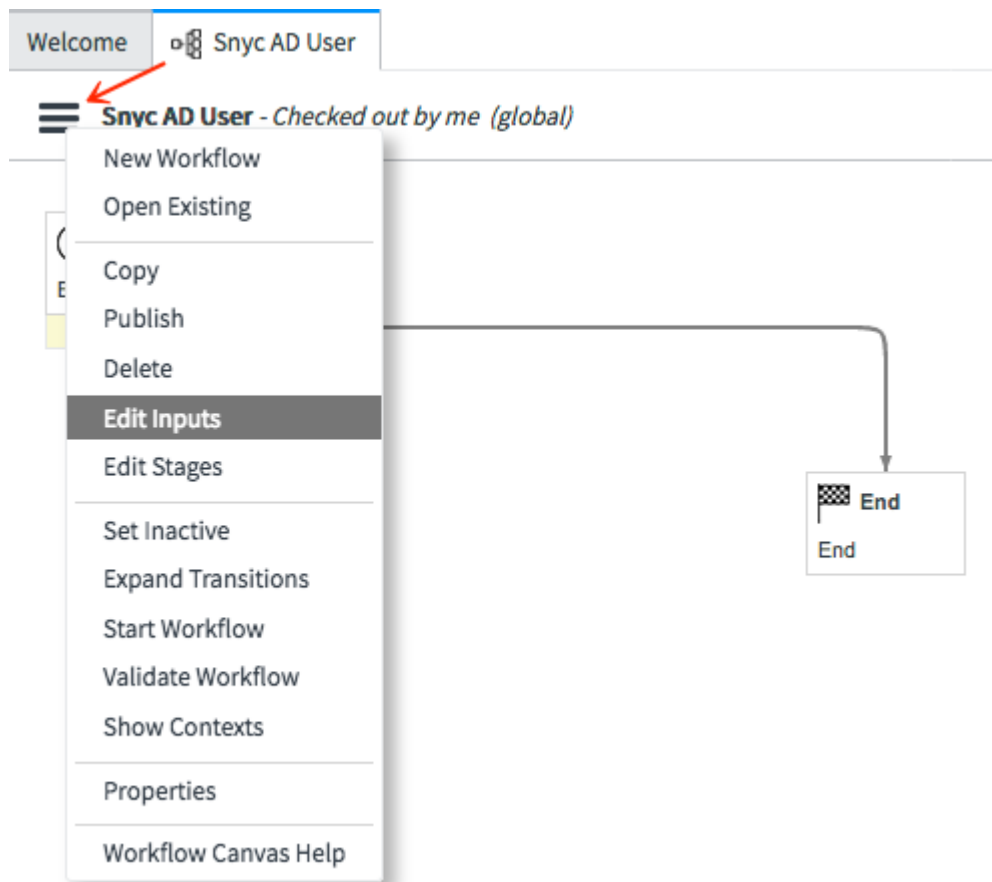
- Name: Sync AD User
- Table: Global [global]



Click **Submit**.

A basic workflow with a Begin and End point appears on the canvas.

Click the menu icon in the upper left corner of the canvas and select **Edit Inputs** from the context menu.



In the Workflow Inputs form, click **New** in the **Variables** record list, and create a new

Field	Value
Type	Reference
Label	User
Column name	u_user
Reference	User

variable, using the fields in the table.

Click **Submit**.

In the **Custom** tab, expand **Custom Activities > Active Directory**.

Drag and drop the Update AD Object activity onto the transition line between the Begin and End points of the new workflow.

This action automatically links the activity with the end point and opens the Workflow Activity property form.

Complete the form using the fields in the table.

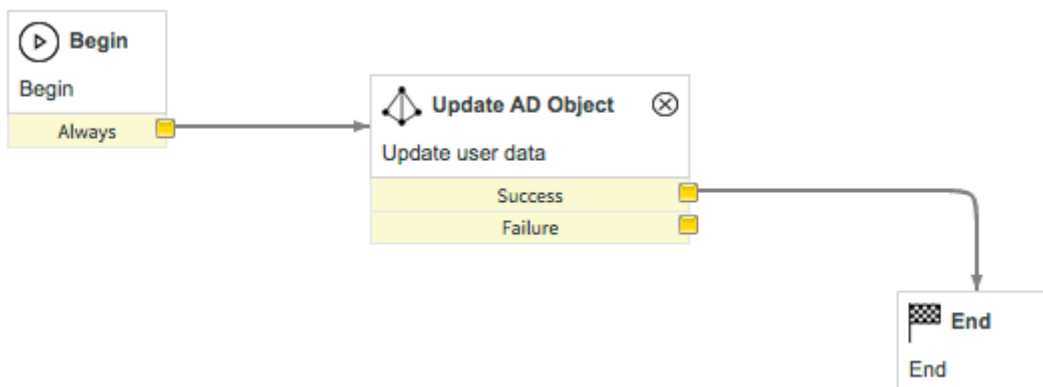
Field	Value
Name	Enter a logical name such as Update user data .



Field	Value
Domain controller	The ServiceNow LDAP integration adds a reference to the LDAP server to which the user's account is linked. To identify the LDAP server, enter the following statement: <code>\${workflow.inputs.u_user.ldap_server.server_url}</code>
Type	Type of AD object. In this case, the type is User , which is the default.
Object name	This example assumes that the ServiceNow user name matches the Active Directory sAMAccountName. Enter the following: <code>\${workflow.inputs.u_user.user_name}</code>
Object data	Updates the user's account in Active Directory, if the user exists. In this example, the user's title is updated: <code>{"title" : "QA"}</code>

Click **Submit**.

The workflow looks like this:



Attach both activity outcomes (Success and Failure) to the end point.

At this point, the workflow takes a ServiceNow user record as input and updates the First Name, Last Name, and Title of the corresponding Active Directory account. If the account does not exist in Active Directory, the workflow fails.

Note: In a normal workflow, some type of alternate action is desirable upon failure. For example, you might send an **email notification** if the workflow failed to update the record.

To prevent the workflow from failing, add a Create AD Object activity to the transition lines between Begin and the Update AD Object activity.

Complete the Workflow Activity property form using the fields in the table.

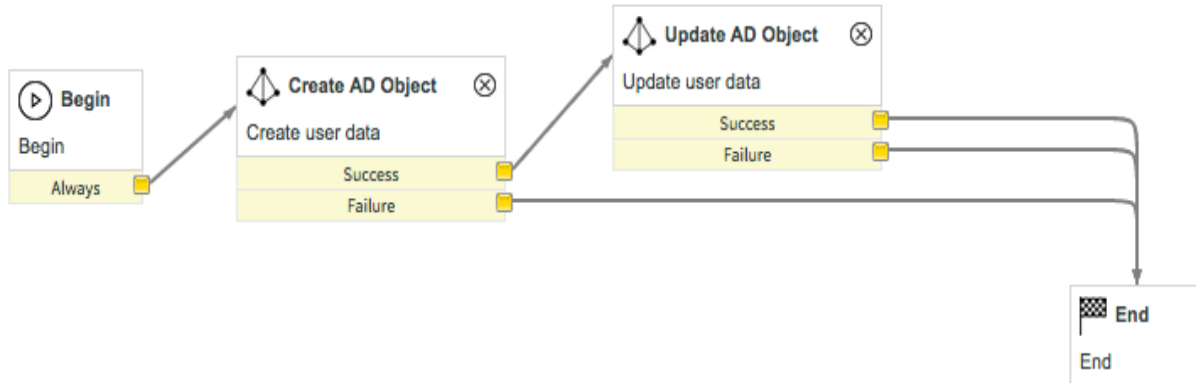
Field	Value
Name	Enter a logical name such as Create user data .
Domain controller	Same as for the update activity. <code>\${workflow.inputs.u_user.ldap_server.server_url}</code>
Ou	The organizational unit to which this object belongs. For the purpose of this example, you can enter OU=HQ,OU=Managed Objects
Object name	Same as for the update activity. <code>\${workflow.inputs.u_user.user_name}</code>
Object data	Creates an account with only the user name in it. For the purpose of this example, you can enter <code>{"givenName" : "\${workflow.inputs.u_user.first_name}", "SN" : "\${workflow.inputs.u_user.last_name}" }</code>



Click **Submit**.

Connect the Failure outcome of the Create AD Object activity to End.

For this example, we are ignoring errors. The workflow now looks like this



This procedure builds a simple workflow that creates a bare-bones Active Directory account consisting of a user name only. The workflow then updates that account with additional information provided by the ServiceNow User [sys_user] table. However, we do not want to execute the Create AD Object activity if the user account already exists. The best practice is to query Active Directory for matching user records and then branch the workflow based on the results of that query. If an account already exists, then the workflow updates the account. If the account does not exist, then the workflow creates the account in Active Directory.

Drag and drop the Query AD activity onto the transition between Begin and Create AD Object.

Complete the Workflow Activity property form using the fields in the table.

Field	Definition
Name	Enter a logical name such as Search for existing account .



Field	Definition
Domain controller	<code>\${workflow.inputs.u_user ldap_server.server_url}</code>
Properties	A comma-separated list of Active Directory properties to return. For example, givenName, SN, title. If the parameter field is blank, then all properties are returned. In this workflow, we leave the field blank.
Search filter	An LDAP filter string that defines the search parameters. Use any valid LDAP filtering criteria. To find user accounts matching the input record, we use: <code>(samaccountname=\${workflow.inputs.u_user.user_name})</code>

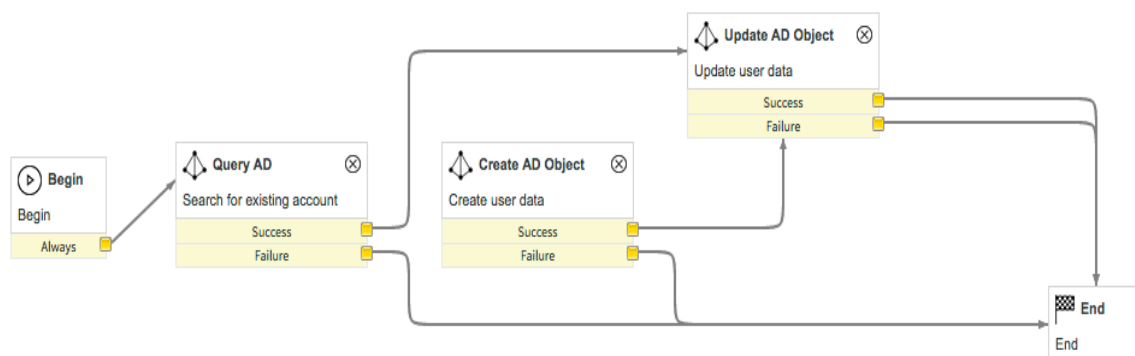
Click **Submit**.

Connect the Failure outcome for the query activity to the End point.

Remember that we are ignoring errors in this workflow.

Connect the Success outcome of the query activity to the Update AD Object activity.

The workflow now looks like this:



The Query AD activity returns its results as a JSON string in the workflow **data bus**. This JSON



string is always an array of objects. Each object corresponds to an Active Directory entry that matched the query. Our workflow should branch, whether that array is empty or not.

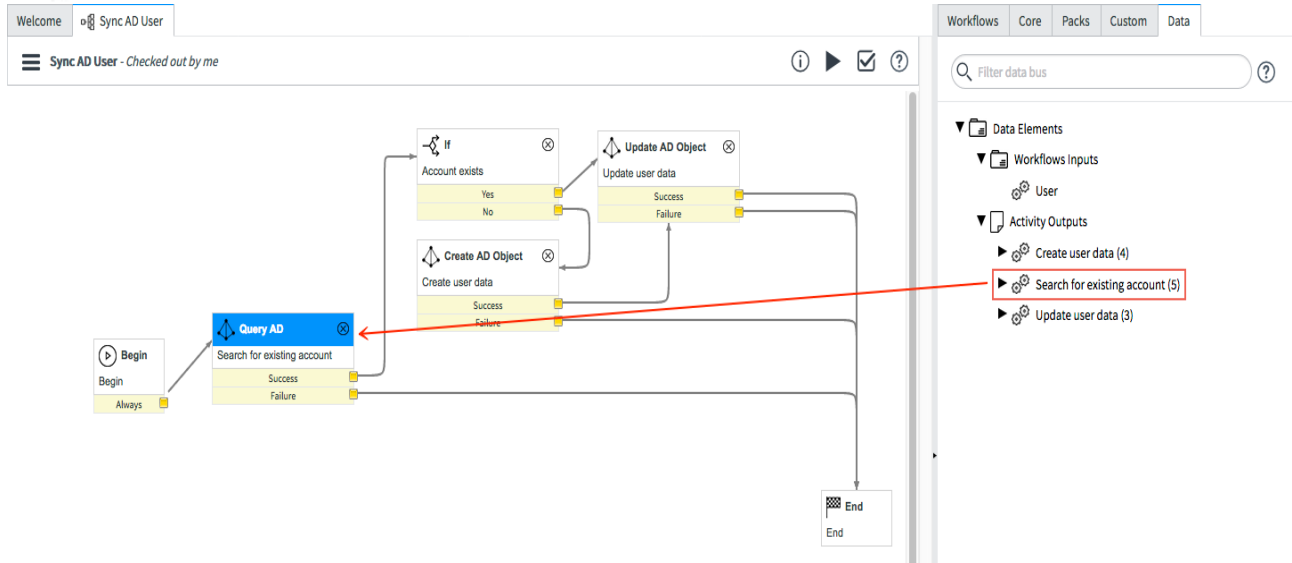
Drag a standard If activity from the Conditions folder in the **Core** tab and drop it onto the transition between Query AD and Update AD Object.

Complete the Workflow Activity properties form using the fields in the table.

Field	Value
Name	Enter a logical name such as Account exists .
Advanced	Select this check box to open the Script field.
Script	<p>For the If activity to work correctly, we must return a yes or no in the answer variable (var) which corresponds to the Yes and No outcomes of the query activity. Line 1 converts the Query AD results from a JSON string into a Javascript array called queryResults. Line 2 checks the length of that array. If the array is more than 0, a match to the account was found, and we set our answer to yes. Otherwise, the answer is no.</p> <pre>var queryResults=new JSON().decode(data.get(5).output); answer = (queryResults.length>0 ? 'yes' : 'no');</pre>

Note: The expression `data.get(5)` in this script identifies the Query AD output in the databus by order number, since it was the fifth activity added to the workflow.





Click **Submit**.

In the canvas, create a transition from the Yes outcome of the If activity to the Update AD Object activity.

Create a transition from the No outcome of the If activity to the Create AD Object activity.

This is the final step. This workflow will query Active Directory to determine if an account already exists. If an account exists, the workflow updates that account. If an account does not exist, the workflow creates the account and then updates the Active Directory with the configured user data set.

