

Okta Orchestration Activity Pack – Install Guide

Overview

The Okta Activity Pack for ServiceNow Orchestration provides ServiceNow administrators with a range of automation options for their Okta infrastructure. The included orchestration workflow activities allows you to integrate Okta controls and activity into your business workflows inside of ServiceNow to provide control over IT operations. By embedding Okta controls into ServiceNow, users can request access and applications from the familiar ServiceNow Service Catalog allowing IT to seamlessly and automatically respond inside of the ServiceNow framework utilizing approvals, documentation, and audit controls. The published workflow activities allow you to add Okta API logic with little to no actual coding.

Example workflows are included and described in this document based on common customer scenarios, but the real power of the activity pack is using the orchestration activities in your own workflows, giving you full control and power of integrating with Okta. Administrators should explore creative ways to leverage Okta inside of existing and future workflows by leveraging the included control points (and future expansions).

Leveraging this app allows Okta to become the application entitlement and enforcement layer in a service-oriented IT architecture. ServiceNow serves as the central control hub for user and IT behaviors, while Okta increases security, enforces application access and permissions, and increases user productivity.

Example solutions leveraging the Activity Pack

The Okta Activity Pack gives ServiceNow administrators control over Groups and Users in Okta. Okta master groups are the primary method for Okta to assign Applications to Users, so control at this layer is extremely powerful. Joint Okta + ServiceNow customers have told us that there are several use-cases that this level of control can help with. Among them are:

- New Hire Onboarding

Workflows in ServiceNow to handle new-hire assignment tasks can now be expanded to include Application Provisioning by way of Okta. IT asset requests like laptops, facilities requests like badges, and application access requests like Salesforce for sales staff can now all be batched together in a single Workflow. Okta administrators can establish a set of roles matching to a set of Groups in Okta, which the Okta Activity Pack can make available to the ServiceNow Workflow.

- Job changes

This use-case is essentially the same as the new hire onboarding use case. A job change will likely require a new set of applications. Workflows in ServiceNow can be constructed to manipulate a user's Group memberships in Okta, thus changing their application access levels.

- Self-Service Group Membership / Self-Service App Requests

Group Memberships in Okta can be directly exposed to Catalog Items in ServiceNow. A user could request membership to a group as part of a special project or time-bound permissions change. Alternately, Administrators could use direct Group Membership Activities to support user self-service App Requests, where a Catalog Item is exposed to users to request a given app, and the outcome of that request is to kick off a Group Add Activity (or Group Remove activity for App unassignment). By tracking these requests and fulfillments in

ServiceNow, audit trail and compliance tracking are far easier, and organizations benefit from self-service work.

- User suspend

ServiceNow is often called upon to serve as the central repository for user decisions. Perhaps a user is traveling to a developing country with a high risk profile. ServiceNow workflows can be leveraged to Suspend that Users application access via standard Okta Lifecycle operations, to protect sensitive company assets. Unsuspending that user when they return to normal operation is equally available. Being able to perform all of these tasks from one place within ServiceNow gives company administrators fast, easy control over risk and compliance requirements.

All of these scenarios are made possible with supported activities that can be used in any workflow so administrators don't have to worry about build scripts to directly call the Okta API methods.

Okta Activity Pack Functional Description

Understanding the flow of data inside of the Activity Pack functionality will help when making architectural decisions. The Okta Activity Pack exposes a set of custom Activities into the ServiceNow Orchestration module. These Activities have a built-in data bus to exchange information between ServiceNow and Okta, to allow for Okta Group and User manipulation from a Workflow in ServiceNow.

In order for the workflow activities to communicate with the Okta APIs there are a few dependencies that need to be in place to support all supported operations.

- There needs to be a consistent mapping of ServiceNow users (sys_user) and Okta users. This mapping is accomplished through a new field (x_okta2_actpack_okta_id) added to the sys_user table. Once synchronized with Okta using the methods defined below this field will store the associated user's Okta ID so it can be leveraged for all activity performed on that user object. By default, user lookups from ServiceNow users to Okta users is performed using the email address value.
- In order to perform group related changes in Okta, ServiceNow needs to be aware of the Okta master groups available along with their Okta ID.

1.! Installation Prerequisites

- You must have ServiceNow Orchestration plugin installed before installing this activity pack.
 - https://docs.servicenow.com/bundle/istanbul-it-operations-management/page/product/orchestration/task/t_ActivateOrchestration.html
- Get the app from the store, this enables it to be installed in your company instances
 - https://store.servicenow.com/sn_appstore_store.do#!/store/application/78d76bbd0f1d2600299f06ace1050e20
- Once your company has activated the application from the store it will be available to install on your registered company instances.
- From the ServiceNow instance where you would like to install the Okta Activity Pack, open the System Applications -> Applications module. From the Downloads tab, install the Okta application.

2.! Configure

After obtaining the Okta Activity Pack from the ServiceNow store, some basic setup is required.

1. Obtain an Okta API token
 - a. Log into Okta as a Super-Administrator
 - b. Navigate to the Admin app (Admin button in top-right of the main UI)
 - c. Navigate to Security-API
 - d. Generate a new Token
 - i. Provide a name for the token, name is for reference only
 - ii. Click Create Token

- iii. Copy the token to Notepad (or similar). You will not be able to retrieve this token once you click OK. If you lose the token you will have to request a new one.
2. From ServiceNow logged in as an admin role user, open the Okta Activity Pack -> Okta Properties module
 - a. If the form is not editable and you see the message:
"This record is in the Okta Orchestration Activity Pack application, but Global is the current application. To edit this record click here." just click the "here" link. This is related to the Okta application being in it's own protected application scope which is standard practice for all ServiceNow certified applications.
 - b.
 3. Update the property fields:
 - a. Okta API token: paste the token value you copied from Okta token creation
 - b. Okta API version: leave the default value of "v1" unless told otherwise by Okta support
 - c. Okta organization URL: the URL of your Okta instance, ex. <https://company.okta.com>
 - d. Okta filter for importing groups: it is recommended that you leave the default filter value. The filter

is used during the import of Okta master groups. Incorrect values will directly impact the functionality related to Okta group activities.

- e. Days to disable Okta group records: In order to honor the propagation of Okta master groups removals the ServiceNow representation of the Okta group will be disabled if the group is not seen for these number of days of automated scheduled imports. It is recommended that you use the default value of "2".
- f. Logging level: Default value is "info" which will capture minimal transaction event data in the ServiceNow application log. For more detailed troubleshooting you can set the value to "debug". Note that this will cause a large amount of logging activity for all Okta activity pack transactions. Use with caution in production instances and return the value to "info" as soon as troubleshooting is complete.

The screenshot shows the ServiceNow 'Okta Properties' configuration page. The page has a dark header with the 'servicenow' logo and 'Service Management' text. Below the header, there's a search bar with 'okta' entered. The main content area is divided into a left sidebar and a right main panel. The sidebar contains links like 'Okta Activity Pack', 'Okta Activity Pack Properties', 'Okta Group Import Service', 'Contact Okta Support', and 'Examples'. The main panel displays the 'Okta Properties' form with fields for 'Okta API token' (masked with asterisks), 'Okta organization instance URL', 'Okta filter to use when importing groups' (with a default value), and 'Days to wait before an Okta group record is disabled if it's not seen during import' (set to 2). There's also a 'Logging level for Okta Activity Pack' dropdown set to 'info' and a 'Save' button at the bottom.

3. Data Dependencies

As mentioned above, the activity pack depends on Okta ID data for both users and groups. In order to prevent any conflicts with the use of ServiceNow group records (sys_user_group), the activity pack creates a new table to store data about the Okta master groups meeting the criteria of the filter defined on the property page. The following defines the details of these Okta metadata imported and stored in ServiceNow.

Okta Master Groups

A new table (x_okta2_actpack_okta_group) has been provisioned to store the required Okta master group metadata. You can access the table from the Okta Activity Pack -> Okta Groups module. This table only lists groups that are mastered by Okta, and not groups that are mastered from Active Directory or other sources such

as O365 or Box. The expectation is that an Okta-mastered group in Okta will drive application or policy assignment from ServiceNow.

The following components are used to populate this table:

- Okta Import Groups Scheduled Job – by default the job runs daily at 6am of the time zone of the instance. You can modify the frequency by navigating to System Definitions -> Scheduled Jobs module and searching for the job name “Okta Import Groups”. You can also manually execute the job at anytime from the job form using the “Execute Now” form button.
- Okta Group Import service staging table (x_okta2_actpack_okta_group_import) – the schedule job populates the import set staging table for processing by the transform map
- Okta Group Import transform map – standard ServiceNow import set transform map to manage the creation and update of target Okta Group records (x_okta2_actpack_okta_group).

User Attribute

The sys_user table will now have a new field (x_okta2_actpack_okta_id) that is used to store the user’s Okta ID. The ID needs to be populated before a user can be used in Okta activities. In the example catalog items, users without an Okta ID value will be filtered out and not be selectable from in the catalog variables. It is recommended that if you build new catalog items or leverage Okta activates in other ways, you ensure that only users with a populated “x_okta2_actpack_okta_id” value be used.

4.1 Populating Required Data from Okta

Initial Okta Master Group [Import](#)

The scheduled job “Okta Import Groups” is scheduled to run daily at 6am. You should run this job manually once the application properties are configured.

- Open the System Definition -> Scheduled Jobs module list
- Search for the Okta Import Groups record
- Click “Execute Now” to start importing groups
- After a few minutes, open the Okta Groups module and confirm group records were created

Adding Okta IDs to ServiceNow User Accounts

By default, there is a business rule “Get Okta ID” on the sys_user table that will attempt to query Okta for a user’s ID anytime the user record is updated, has an email value, and does not yet contain an Okta ID field value. This field will not be displayed on the user table list or form views. You can add this manually if you wish to review the values.

The query to locate the associated user in Okta is [profile.email eq ‘servicenow user email’]. It is recommended that you do not modify this rule. If it does not fit your data model then you can disable this rule and create your own to query Okta.

Since this rule is only executed when a user record is inserted or updated you may need to perform a bulk update operation to get started. You can execute the scheduled job “Okta Bulk User ID Update” to query all users that

have an email value and do not have an Okta ID value already in ServiceNow. This scheduled job is not scheduled to run and should only be used for on-demand execution as needed.

Warning: the API is limited to 600 user queries at a time. If you have more than 500 users that need updating you should contact Okta support for a different script.

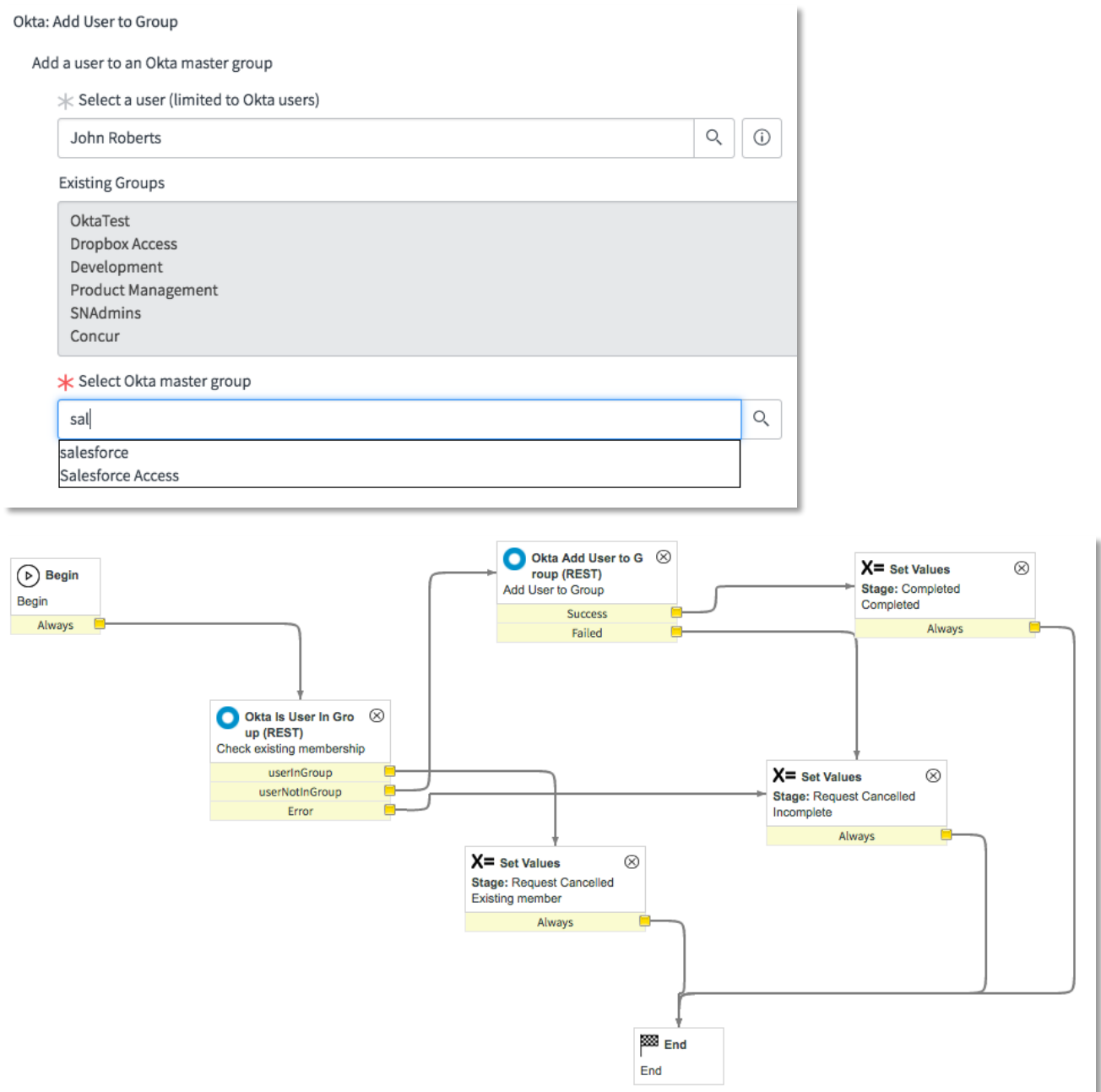
5. Usage Requirements

- The included example catalog items as well as the orchestration activities require that ServiceNow users (sys_user) and Okta Groups (x_okta2_actpack_okta_group) have Okta ID field populated with the ID's from the matching Okta record
- You will need to add the Okta ID to the user (sys_user) form and list if you want to view/confirm the values in ServiceNow. This field is read-only by default.
- Any user Create or Update action in ServiceNow will trigger a query to Okta to find an Okta using matching the email on the ServiceNow user record. Only users with an Okta ID will be able to be managed with the orchestration activities.
 - Note that email must match for this query to succeed. There may be a situation where Okta Usernames are not in an email format. In this case, contact Okta for assistance as this may require Professional Services assistance.
 - Note that some operations in ServiceNow can trigger an Update event on every record in the sys_user table, which could potentially cause a large volume of calls to Okta. It is recommended that Okta be used to provision users, rather than bulk .CSV or LDAP updates from separate sources.

6. Using the Example Catalog Items and Workflows

The activity pack includes the following example catalog items and workflows to demonstrate the functionality available using the Okta activities. You can use these examples as-is, modify them, or build your own using the Okta activities in your own workflows. Here is a detailed review of each workflow.

- **Okta: Add User to Group** - request and orchestration to add an Okta user (linked to a ServiceNow user) to an Okta master group



- Variables:
 - user – reference to sys_user limited to users with Okta ID, defaults to current session user
 - group – reference to x_okta2_actpack_okta_group limited to active groups
 - existing_groups and existing_group_ids – track and display the list of groups that a user
- Client Scripts:
 - Get Okta User's Groups – on change of the user variable the client script queries Okta for the

list of existing groups for that user. The group names will be displayed on the page and existing groups will be removed from the 'group' variable choices.

- Workflow: Okta Add User to Group
 - Activity: Okta Is User in Group – takes Okta IDs for the Okta group and user, checks Okta for existing group membership
 - Activity: Add user to Group – takes Okta IDs for the Okta group and user, adds the user to the Okta group

- **Okta: Remove User from Group** - request and orchestration to remove an Okta user (linked to a ServiceNow user) from an Okta master group

- Variables:

- user – reference to sys_user limited to users with Okta ID, defaults to current session user
- group – reference to x_okta2_actpack_okta_group limited to active groups, limited to the

- **Okta: User Lifecycle** - request and orchestration to perform an Okta lifecycle action on an Okta user (linked to a ServiceNow user)

Okta: User Lifecycle Management

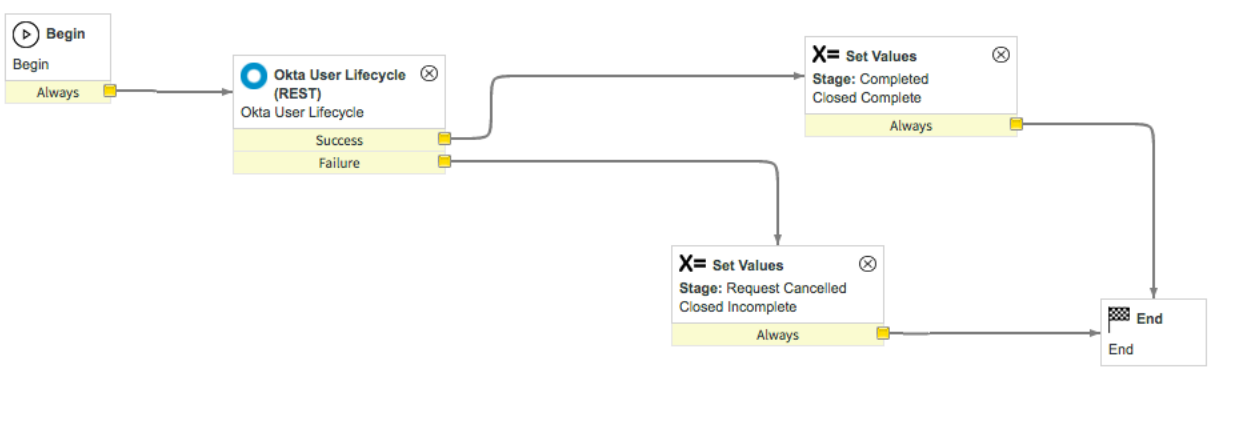
Lifecycle operations for Okta users

* User

Sally Test

* Operation

Suspend



- Variables:
 - user – reference to sys_user limited to users with Okta ID, defaults to current session user
 - operation – supports Okta user lifecycle operations (see Okta API documentation for more info), common operations are activate, deactivate, suspend, unsuspend, unlock, reset password
- Client Scripts:
 - None
- Workflow: Okta User Lifecycle
 - Activity: Okta User Lifecycle – takes Okta user ID and operation name, attempts to perform operation on the Okta user

7.1 Available Okta Activities

From any workflow you'll be able to access the available Okta activities under the "Packs" tab. You can also view the published inputs and outputs for each activity.

The screenshot displays the 'Okta Activities' interface. On the left, a search bar contains the text 'Filter orchestration packs'. Below it, a tree view shows the following structure:

- Orchestration Packs
 - Okta Inc
 - Okta Orchestration Activity Pack
 - Okta Add User to Group (REST) - v1
 - Okta Is User In Group (REST) - v1
 - Okta Remove User from Group (REST) - v1
 - Okta User Lifecycle (REST) - v1
 - ServiceNow

On the right, the details for the 'Okta Add User to Group (REST) - v1' activity are shown:

- Okta Add User to Group (REST) - v1
 - Inputs
 - group_id
 - user_id
 - Outputs
 - status_code
 - error

Activity details

All Okta activities are set to read-only mode. If you need to modify the core functionality you can create new custom activity and manually copy the specific configurations and scripts from the Okta provided activities.

Output values are available to other activities using the workflow data bus values. See ServiceNow workflow documentation for more information.

- **Okta Add User to Group** - add Okta user to an Okta master group
 - Related Okta API: <https://developer.okta.com/docs/api/resources/groups.html#add-user-to-group>
 - (input) group_id: Okta group id, can be retrieved from imported okta groups, okta_id field
 - (input) user_id: Okta user id, can be retrieved from sys_user.x_okta2_actpack_okta_id field
 - (output) status_code: Okta API request status code
 - (output) error: any error from Okta API request
 - (condition) Success: status_code == 204
 - (condition) Failed: (else condition)
- **Okta Is User In Group** - check to see if an Okta user is a member of an Okta group
 - Related Okta API: <https://developer.okta.com/docs/api/resources/groups.html#list-group-members>
 - (input) group_id: Okta group id, can be retrieved from imported okta groups, okta_id field
 - (input) user_id: Okta user id, can be retrieved from sys_user.x_okta2_actpack_okta_id field
 - (output) userInGroup: true or false, is user in the group
 - (output) status_code: Okta API request status code
 - (output) error: any error from Okta API request
 - (condition) userInGroup: userInGroup == "true"
 - (condition) userNotInGroup: userInGroup == "false"
 - (condition) Error: (else condition)
- **Okta Remove User from Group** - remove Okta user from Okta master group
 - Related Okta API: <https://developer.okta.com/docs/api/resources/groups.html#remove-user-from-group>
 - (input) group_id: Okta group id, can be retrieved from imported okta groups, okta_id field
 - (input) user_id: Okta user id, can be retrieved from sys_user.x_okta2_actpack_okta_id field
 - (output) status_code: Okta API request status code
 - (output) error: any error from Okta API request
 - (condition) Success: status_code == 204
 - (condition) Failed: (else condition)
- **Okta User Lifecycle** - perform a user lifecycle action on an Okta user
 - Related Okta API: <https://developer.okta.com/docs/api/resources/users.html#lifecycle-operations>
 - (input) user_id: Okta user id, can be retrieved from sys_user.x_okta2_actpack_okta_id field
 - (input) operation: lifecycle operation (see Okta API)
 - (input) send_email: *optional*, true/false on whether the action should send the user an email from Okta
 - (input) query_parm: *optional* query parameters (see Okta API)
 - (output) response: API request response body
 - (output) status_code: Okta API request status code

- (output) error: any error from Okta API request
- (condition) Success: status_code = 200
- (condition) Failed: Not Success

8.1 Troubleshooting

Simple test for validating API url, token and access

- Run the following script from the Scripts – Background module

```
var okta = new x_okta2_actpack.OktaRESTActivity();
var rest = okta.oktaRESTRequest("get", "users");
```
- You should get a log statement confirming "oktaRESTRequest response status:200" this means the API test is successful. If status is not 200, review the output for additional information.
- If there is a problem with the API token you will get the following response:
 - **response status:401**, body:{"errorCode":"E0000011","errorSummary":"Invalid token provided"}

Detailed Debug Logging

- From the Okta properties page, set the logging level to 'debug'
- Review the System Logs -> Application Logs lists for debug details
- Review the System Logs -> All lists for debug details

9. Application Component List – Supplemental

This information is provided as a reference guide for detailed ServiceNow architects. No modifications or configuration steps should be required for normal use.

Data Model

New Tables

Okta Group Import (x_okta2_actpack_okta_group_import)

- Extended from: sys_import_set_row
- Extensible: No
- Application Access
 - Access from:
 - Via web services: Yes
 - Allow configuration: No
 - Operations: Read
- Columns
 - Description (description) - String
 - ID (id) - String
 - Name (name) - String
 - Type (type) - String
- ACL Count

	Read	Write	Create	Delete
Row	1	1	1	1
Column	1	1	1	n/a

Okta Group (x_okta2_actpack_okta_group)

- Extended from: N/A
- Extensible: No
- Application Access
 - Access from: All application scopes
 - Via web services: Yes
 - Allow configuration: No
 - Operations: Read
- Columns
 - Active (active) - True/False
 - Description (description) - String
 - Last import (last_import) - Date/Time
 - Name (name) - String
 - Okta ID (okta_id) - String
 - Type (type) - String
- ACL Count

	Read	Write	Create	Delete
Row	1	1	1	1
Column	1	1	1	n/a

New Columns

Okta ID (sys_user.x_okta2_actpack_okta_id) - String

- ACL Count

Read	Write	Create
1	1	0

Server Development

Business Rules

- Get Okta ID sys_user (async: insert,update)

Script Includes

- OktaUtils
 - Description: var otkautils = new x_okta2_actpack.Okta_Utils(); otkautils.importGroups();
 - Access from: All application scopes
- OktaAJAX
 - Description: AJAX functions to Okta related client scripts
 - Access from: All application scopes
 - Client callable script: has ACL = Yes
- OktaRESTActivity
 - Description: Okta REST wrappers for use in scripts or orchestration activities
 - Access from: All application scopes

Scheduled Jobs

- Okta Import Groups
 - Daily - 06:00:00
- Okta Bulk User ID Update
 - On Demand - 00:00:00

Access Control

Roles

- x_okta2_actpack.admin: Okta Activity Pack administrator role
- x_okta2_actpack.user: User level access to Okta activity pack components

Access Controls

- OktaAJAX client_callable_script_include:execute (Cond: No, Script: No, Roles: x_okta2_actpack.admin,x_okta2_actpack.user)
- sys_user.x_okta2_actpack_okta_id record:read (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- sys_user.x_okta2_actpack_okta_id record:write (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group record:create (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group record:delete (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group record:read (Cond: No, Script: No, Roles: x_okta2_actpack.user)
- x_okta2_actpack_okta_group record:write (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group.* record:create (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group.* record:read (Cond: No, Script: No, Roles: x_okta2_actpack.user)
- x_okta2_actpack_okta_group.* record:write (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group_import record:create (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group_import record:delete (Cond: No, Script: No, Roles: x_okta2_actpack.admin)

- x_okta2_actpack_okta_group_import record:read (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group_import record:write (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group_import.* record:create (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group_import.* record:read (Cond: No, Script: No, Roles: x_okta2_actpack.admin)
- x_okta2_actpack_okta_group_import.* record:write (Cond: No, Script: No, Roles: x_okta2_actpack.admin)

Properties

- x_okta2_actpack.api_token=: Okta API token, obtain the API token from Okta Security-API page (password2)
 - Read: x_okta2_actpack.admin
 - Write: x_okta2_actpack.admin
- x_okta2_actpack.api_version=v1: Okta API version (string)
 - Read: x_okta2_actpack.admin
 - Write: x_okta2_actpack.admin
- x_okta2_actpack.deactivate_old_groups_days=2: How many days to wait before an Okta group record is disabled if it's not seen during import (integer)
 - Read: x_okta2_actpack.admin
 - Write: x_okta2_actpack.admin
- x_okta2_actpack.group_import_filter=type+eq+%22OKTA_GROUP%22: Okta filter to use when importing groups (default=type+eq+%22OKTA_GROUP%22) (string)
 - Read: x_okta2_actpack.admin
 - Write: x_okta2_actpack.admin
- x_okta2_actpack.logging.verbosity=info: Logging level for Okta Activity Pack (choice list)
 - Read: x_okta2_actpack.admin
 - Write: x_okta2_actpack.admin
- x_okta2_actpack.org_instance_url=: Okta organization instance URL (string)
 - Read: x_okta2_actpack.admin
 - Write: x_okta2_actpack.admin

Navigation

Menu and Modules (desktop)

- Okta Activity Pack / Okta Properties: URL (from Arguments:):
- Okta Activity Pack / Okta Groups: List of Records: x_okta2_actpack_okta_group
- Okta Activity Pack / Okta Group Import Service: URL (from Arguments:):
- Okta Activity Pack / Contact Okta Support: URL (from Arguments:):
- Okta Activity Pack / Examples: Separator
- Okta Activity Pack / Catalog Items: List of Records: sc_cat_item

Service Catalog

Catalog Client Scripts

- Get Okta User's Groups (onChange)
 - Description:
 - Applies to: item: Okta: Remove User from Group
- Get Okta User's Groups (onChange)
 - Description:
 - Applies to: item: Okta: Add User to Group

Integrations

Import Transforms

- Okta Group Import
 - Source: x_okta2_actpack_okta_group_import (x_okta2_actpack_okta_group_import)
 - Target: x_okta2_actpack_okta_group (x_okta2_actpack_okta_group)

REST Messages

- Okta Get Groups
 - Description: Example REST Message to GET Okta Groups
 - Access from: All application scopes]
 - Endpoint:
 - Methods: get

Workflow

Activities

- Okta Remove User from Group (REST) Run Script
 - Description: Activity to remove a user from an Okta group
 - Access from: All application scopes
 - Has checkout: false
- Okta User Lifecycle (REST) Run Script
 - Description: Okta User Lifecycle Management Operations
 - Access from: All application scopes
 - Has checkout: false
- Okta Add User to Group (REST) Run Script
 - Description: Activity to add an Okta user to an Okta group
 - Access from: All application scopes
 - Has checkout: false
- Okta Is User In Group (REST) Run Script
 - Description: Verify a user is in a group
 - Access from: All application scopes
 - Has checkout: false

Workflows

- Okta Remove User from Group ()
 - **Has published: true**
 - Has checkout: false
- Okta User Lifecycle (sc_req_item)
 - **Has published: true**
 - Has checkout: false
- Okta Add User to Group (sc_req_item)
 - **Has published: true**
 - Has checkout: false

Other App File Records

- Catalog UI Policy: Existing group fields are read-only (catalog_ui_policy)
- Catalog UI Policy: Existing group fields are read-only (catalog_ui_policy)
- Catalog UI Policy Action: existing_groups (catalog_ui_policy_action)
- Catalog UI Policy Action: existing_groups (catalog_ui_policy_action)
- Catalog UI Policy Action: existing_group_ids (catalog_ui_policy_action)

- Catalog UI Policy Action: existing_group_ids (catalog_ui_policy_action)
- Images: okta_logo_sm.png (db_image)
- Variable: Existing Group IDs (item_option_new)
- Variable: Existing Group IDs (item_option_new)
- Variable: Existing Groups (item_option_new)
- Variable: Existing Groups (item_option_new)
- Variable: Operation (item_option_new)
- Variable: Select a user (limited to Okta users) (item_option_new)
- Variable: Select a user (limited to Okta users) (item_option_new)
- Variable: Select Okta master group (item_option_new)
- Variable: Select Okta master group (item_option_new)
- Variable: User (item_option_new)
- Question Choice: Activate (question_choice)
- Question Choice: Deactivate (question_choice)
- Question Choice: Expire password (question_choice)
- Question Choice: Reset factors (question_choice)
- Question Choice: Reset password (question_choice)
- Question Choice: Suspend (question_choice)
- Question Choice: Unlock (question_choice)
- Question Choice: Unsuspend (question_choice)
- Category: Okta (sc_category)
- Category Available for: Okta.Users with 'x_okta2_actpack.user' role (sc_category_user_criteria_mtom)
- Catalog Item: Okta: Add User to Group (sc_cat_item)
- Catalog Item: Okta: Remove User from Group (sc_cat_item)
- Catalog Item: Okta: User Lifecycle (sc_cat_item)
- Catalog Items Catalog: Service Catalog.Okta: Add User to Group (sc_cat_item_catalog)
- Catalog Items Catalog: Service Catalog.Okta: Remove User from Group (sc_cat_item_catalog)
- Catalog Items Catalog: Service Catalog.Okta: User Lifecycle (sc_cat_item_catalog)
- Catalog Item Category: Okta.Okta: Add User to Group (sc_cat_item_category)
- Catalog Item Category: Okta.Okta: Remove User from Group (sc_cat_item_category)
- Catalog Item Category: Okta.Okta: User Lifecycle (sc_cat_item_category)
- Catalog Item Available for: Okta: Add User to Group.Users with 'x_okta2_actpack.user' role (sc_cat_item_user_criteria_mtom)
- Catalog Item Available for: Okta: Remove User from Group.Users with 'x_okta2_actpack.user' role (sc_cat_item_user_criteria_mtom)
- Catalog Item Available for: Okta: User Lifecycle.Users with 'x_okta2_actpack.user' role (sc_cat_item_user_criteria_mtom)
- System Property Category: Okta Activity Pack (sys_properties_category)
- HTTP Headers: Authorization (sys_rest_message_fn_headers)
- HTTP Headers: Authorization (sys_rest_message_headers)
- HTTP Headers: Content-Type (sys_rest_message_headers)
- Scope Script Access: AbstractAjaxProcessor (sys_scope_script_access)
- Scope Script Access: JSON (sys_scope_script_access)
- Form Section: x_okta2_actpack_okta_group (sys_ui_section)
- Style: x_okta2_actpack_okta_group (sys_ui_style)
- Style: x_okta2_actpack_okta_group (sys_ui_style)
- Contained Role: x_okta2_actpack.user.x_okta2_actpack.admin (sys_user_role_contains)
- User Criteria: Users with 'x_okta2_actpack.user' role (user_criteria)