Administrator Guide

IM.Nextcloud Connector

Autor: Nicolai Kolandjian

Erstelldatum: 30-05-2022

Letzte Änderung: 30-05-2022

Kontrollnummer: OC\_DE/300516442/DO.xxx/xxxx

Version: 1.0

**Freigabe:**

|  |  |
| --- | --- |
| Bundeskriminalamt |  |
| Oracle Consulting |  |

logored Kopie-Nr. \_\_\_\_\_

## Dokumentenkontrolle

### Änderungshistorie

| Date | Autor | Version | | Änderungsreferenz |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
| 30-05-2022 | Nicolai Kolandjian | 1.0 | Initial Document | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |

### Reviewer

| Name | Position |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Inhalt

[Dokumentenkontrolle ii](#_Toc104762696)

[Änderungshistorie ii](#_Toc104762697)

[Reviewer ii](#_Toc104762698)

[Inhalt iii](#_Toc104762699)

[Introduction 1](#_Toc104762700)

[Related Documents 1](#_Toc104762701)

[Vertraulichkeit 1](#_Toc104762702)

[Nextcloud Connector 2](#_Toc104762703)

[Hardware und Software Zertifizierung 2](#_Toc104762704)

[Erforderliche Komponentenversionen 2](#_Toc104762705)

[Erforderliche Patches 2](#_Toc104762706)

[Architecture 2](#_Toc104762707)

[Authorization 3](#_Toc104762708)

[Multiple Targets Support 3](#_Toc104762709)

[Design Notes 4](#_Toc104762710)

[Create User Limitations 4](#_Toc104762711)

[Validating Phone Update Operations 4](#_Toc104762712)

[Duplicated UI Form Fields 4](#_Toc104762713)

[Reconciliation Matching Rules 5](#_Toc104762714)

[Full Reconciliation 5](#_Toc104762715)

[Incremental Reconciliation 5](#_Toc104762716)

[Filtered Reconciliation 5](#_Toc104762717)

[Lookups 5](#_Toc104762718)

[Preconfigured Lookups 6](#_Toc104762719)

[Synchronized Lookups 7](#_Toc104762720)

[Deploying the Connector 9](#_Toc104762721)

[Deploy the ICF Bundle 9](#_Toc104762722)

[Deploy OIM Shared Metadata 9](#_Toc104762723)

[Import XML Files 9](#_Toc104762724)

[Configure UI Form 10](#_Toc104762725)

[Importing the SSL certificate 10](#_Toc104762726)

[Configure OIM Instance Metadata 10](#_Toc104762727)

[Preparation 10](#_Toc104762728)

[Configure the IT Resource 10](#_Toc104762729)

[Configure the Application Instance 11](#_Toc104762730)

[Create Schedule Jobs 12](#_Toc104762731)

[Using the Connector 13](#_Toc104762732)

[Reconciliation Processes 13](#_Toc104762733)

[Group Definition Reconciliation 13](#_Toc104762734)

[User Create Reconciliation 13](#_Toc104762735)

[User Delete Reconciliation 13](#_Toc104762736)

[User Search Create Reconciliation 14](#_Toc104762737)

[Provisioning Operations 14](#_Toc104762738)

[Accounts 14](#_Toc104762739)

[Entitlements 14](#_Toc104762740)

[Extending the functionality of the Connector 15](#_Toc104762741)

[Reconciliation Operations 15](#_Toc104762742)

[reconPageSize 15](#_Toc104762743)

[Provisioning Operations 15](#_Toc104762744)

[postUpdateValidateFields 15](#_Toc104762745)

[Appendix 16](#_Toc104762746)

[Using ICF Filters 16](#_Toc104762747)

[Local Queries 16](#_Toc104762748)

[Remote Queries 16](#_Toc104762749)

[Constructing ICF Filters 16](#_Toc104762750)

[Importing and Trusted the SSL Certificate 16](#_Toc104762751)

[Enabling Logging 17](#_Toc104762752)

[Understanding Logging Levels 18](#_Toc104762753)

[Grundlegendes zu Ebenen der Protokollierung 18](#_Toc104762754)

[Example Raw API User Data Response 19](#_Toc104762755)

## Introduction

This document related to the installation of the Nextcloud connector, which is an Oracle Identity Manager (OIM) component within the Oracle Identity and Access Management infrastructure.

### Related Documents

For more information about the Nextcloud API’s, refer to the following:

* <https://docs.nextcloud.com/server/latest/admin_manual/configuration_user/user_provisioning_api.html>

### Vertraulichkeit

Das in dieser Dokumentation enthaltene Material enthält geschützte, vertrauliche Informationen zu Oracle-Produkten und –Methoden.

Die Leserschaft erklärt sich damit einverstanden, dass die Informationen in dieser Dokumentation nicht nach außerhalb weitergegeben und nicht für andere Zwecke als zur Bewertung dieses Verfahrens vervielfältigt, verwendet oder weitergegeben werden.

## Nextcloud Connector

Oracle Identity Manager (OIM) requires connectors in order to manage identity data within external systems. In this case the Nextcloud connector enables OIM to manage identity data within Nextcloud using Nextcloud’s REST API’s.

OIM connectors will typically integrate with other systems‘ Identity data, either as a “Trusted“ source, or as a “Target“ resource. This custom Nextcloud connector only supports the integration as a Target resource.

### Hardware und Software Zertifizierung

Die plattformspezifischen Anforderungen an Hardware und Software, die in diesem Dokument aufgeführt werden, sind gültig für den Zeitpunkt zu dem, dieses Dokument erstellt wurde. Da neue Plattformen und Betriebssysteme zertifiziert werden können, nachdem dieses Dokument veröffentlicht wurde, wird empfohlen die Zertifizierungsmatrix auf Oracle Technology Network heranzuziehen. Dort befinden sich die aktuellsten Aussagen zu zertifizierten Plattformen und Betriebssystemen.

Die jeweilige Zertifizierungsmatrix für Produkte der Oracle Identity und Access Management Suite sind unter folgenden URLs verfügbar:

[Oracle® Fusion Middleware 12c (12.2.1.3.0)](https://www.oracle.com/technetwork/middleware/fmw-122130-certmatrix-3867828.xlsx)

#### Erforderliche Komponentenversionen

| **Komponente** | **Version** |
| --- | --- |
| **Oracle Java Development Kit** | JDK 1.8.0\_131 oder höher |
| **Oracle Infrastruktur** | Oracle® WebLogic 12c (12.2.1.3.0) |
| **Oracle Datenbank** | Oracle® RDBMS 12c (12.2.0.1.0 oder höher |
| **Oracle Identity Governance** | Oracle® Identity Governance 12c Release 12.2.1.3.0 |

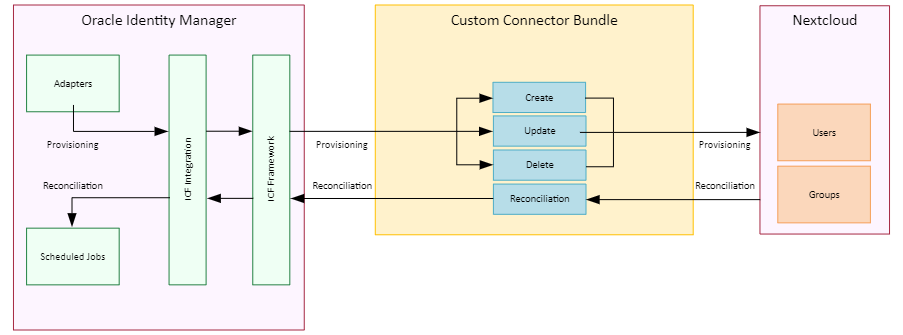
#### Erforderliche Patches

| **Komponente** | **Patch** |
| --- | --- |
| **Oracle Identity Governance** | Patch 33470976 (IDM Suite Bundle Patch ID:12.2.1.3.211014) = Oracle® Identity Governance Bundle Patch 33420133 ID:12.2.1.3.211001 |

### Architecture

The Nextcloud Connector is implemented using the Identity Connector Framework (ICF) which acts as a Service Provider Interface (SPI) to the custom Nextcloud Connector bundle code.

ICF is a component that provides basic reconciliation and provisioning operations that are common to all Oracle Identity Manager connectors. In addition, ICF offers general functions that developers would otherwise have to implement themselves, e.g. Connection pooling, buffering, timeouts and filtering. The ICF ships with Oracle Identity Manager.



This diagram depicts how the connector bundle is responsible for interacting with Nextcloud. The outbound Provisioning and Inbound Reconciliation operations rely on Nextcloud’s HTTP REST API’s.

### Authorization

The Nextcloud REST API requires Authorization on every HTTP request in the form of account credentials provided as an Authorization header (Basic Authentication).

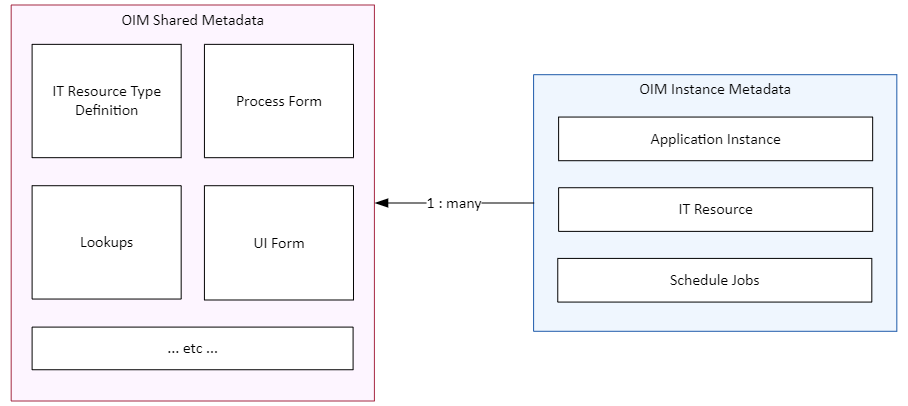
Therefore a service account is required to be provided to OIM with the relevant permissions. For example the "Admins" group is sufficient in a default Nextcloud installation.

### Multiple Targets Support

This custom connector is ready to quickly deploy as one instance.

However it will support using multiple target instances, on the limitation that they will share some of the same metadata. For example, they will all share they same form fields.

| **Note** |
| --- |
| The development of this custom connector could be further expanded upon in order to remove this "limitation" by extracting some or all elements of the shared metadata, and converting them to instance metadata. |



This diagram illustrates how the metadata is divided between:

* Shared Metadata
* Instance Metadata

Therefore in order to create multiple target instances of Nextcloud within OIM, simply follow the steps to create the following for each target instance:

* + Configure the Application Instance
  + Configure the IT Resource
  + Create Schedule Jobs

### Design Notes

#### Create User Limitations

The Nextcloud API will gladly accept many attribute values during the create http operation, and will return a false positive. However the problem here is this is somewhat a false-positive. This is because some attributed will be completely ignored by Nextcloud during the create http operation.

Therefore the following Process Tasks are automatically triggered on the event of a successful **Create User** Process Task, because their respective form fields are completely ignored by Nextcloud during **Create User**:

* Display Name Updated
* Phone Updated
* Locale Updated

This means everytime a Nextcloud resource is provisioned to a user in OIM, there will be minimum 4 network requests.

#### Validating Phone Update Operations

When updating (provisioning an update) the phone field, the Nextcloud API will always respond with a success. The problem is that it will sometimes fail to actually update the phone field, and therefore some responses are false-positives.

Therefore a feature of this custom connector has been implemented to allow post-update validation checks by performing a second API request to retrieve user data, and validate the respnse.

Upon failure, the Process Task which is responsible for the update will error, and will show as "Failed" in the Task History.

*See section postUpdateValidateFields for more details.*

#### Duplicated UI Form Fields

The **Unique ID** and **User ID** OIM form fields both represent that same exact field within the target system - **id**. The reason this mapping is duplicated in the UI Form is because OIM has limitations around referring to the same field for both **\_\_UID\_\_** and **\_\_NAME\_\_**.

For inbound reconciliation, both fields will be appropriately mapped. For outbound provisioning operations, **Unique ID** is a display only field, which means it cannot be modified.

#### Reconciliation Matching Rules

The rule upon which OIM users match against NextCloud accounts are:

* OIM.User Login = NextCloud.userid

In this rule element means:

| **element** | **meaning** |
| --- | --- |
| **User Login** | The User ID field of the OIM user |
| **userid** | This is the "id" value of the user within Nextcloud. It’s the equivalent of a unique and distinct login field. |

#### Full Reconciliation

Full reconciliation is achieved by executing the following Schedule Jobs in sequence:

* User Create Reconciliation
* User Delete Reconciliation

#### Incremental Reconciliation

This connector does not support incremental reconciliation. This is due to limitations of the Nextcloud API whereby it does not have the facility to filter against any attributes which can represent "recent changes"

#### Filtered Reconciliation

Restricted or filtered matching limits the number of records that are matched based on specified filter criteria.

This connector supports filtered reconciliation only for the following Schedule Jobs:

* User Search Create Reconciliation

### Lookups

Lookups (also referred to as List of Values) are used during data reconciliation and provisioning. They can be categorised as the follows:

* Preconfigured Lookups
* Synchronized Lookups

#### Preconfigured Lookups

Preconfigured lists of values are created in OIM when you deploy the connector. These lookups are pre-populated with values and whilst they can be optionally modified, this is not expected to be required.

Within this category of value lists, a distinction is made between:

1. Global Lookups
2. Local Lookups

##### Global Lookups

Global Lookups are independent of a specific target system. They would be shared across many

1. Lookup.NextCloud.Configuration
2. Lookup.NextCloud.UM.Configuration
3. Lookup.NextCloud.UM.ProvAttrMap
4. Lookup.NextCloud.UM.ReconAttrMap

###### Lookup.NextCloud.Configuration

| **Encode** | **decoded** | **description** |
| --- | --- | --- |
| **Bundle Version** | 1 |  |
| **Bundle Name** | bka.iam.identity.nextcloud |  |
| **Connector Name** | bka.iam.identity.nextcloud.NextcloudConnector |  |
| **User Configuration Lookup** | Lookup.NextCloud.UM.Configuration |  |
| **postUpdateValidateFields** | Phone |  |
| **reconPageSize** | 1000 |  |
|  |  |  |

###### Lookup.NextCloud.UM.Configuration

| **Encode** | **decoded** | **description** |
| --- | --- | --- |
| **Recon Attribute Map** | Lookup.NextCloud.UM.ReconAttrMap |  |
| **Provisioning Attribute Map** | Lookup.NextCloud.UM.ProvAttrMap |  |
|  |  |  |

###### Lookup.NextCloud.UM.ProvAttrMap

| **Encode** | **decoded** | **description** |
| --- | --- | --- |
| **Unique ID** | \_\_UID\_\_ |  |
| **User ID** | \_\_NAME\_\_ |  |
| **Password** | \_\_PASSWORD\_\_ |  |
| **Email** | email |  |
| **Display Name** | displayname |  |
| **Locale** | locale |  |
| **Language** | language |  |
| **Phone** | phone |  |
| **UD\_NEXT\_GR~Group Name[LOOKUP]** | groups |  |
|  |  |  |

###### Lookup.NextCloud.UM.ReconAttrMap

| **Encode** | **decoded** | **description** |
| --- | --- | --- |
| **Unique ID** | \_\_UID\_\_ |  |
| **userid** | \_\_UID\_\_ |  |
| **Status** | \_\_ENABLE\_\_ |  |
| **email** | Email |  |
| **displayname** | displayname |  |
| **locale** | locale |  |
| **language** | language |  |
| **phone** | phone |  |
| **Groups~name[LOOKUP]** | groups |  |
|  |  |  |

##### Local Lookups

Local lookups are only available within a specific target system (stage).

*None. See section: Multiple Targets Support*

#### Synchronized Lookups

During a provisioning operation, you use a Lookup in the process form to select a single value from a range of values. For example, you might want to select a group from the Groups search box to specify the group that will be assigned to the user.

After the connector is deployed, the following value lists are automatically created in Oracle Identity Manager to be used as a source for value lists:

1. Lookup.NextCloud.Groups

By default, this lookup is empty. It is populated with values retrieved from the target system when you run the Lookup synchronization background process. For example, when you run the background process, all groups on the target system are retrieved by OIM and placed in the Lookup.NextCloud.Groups list of values.

After synchronization, the data in each of the value lists is stored in the following format:

| **value** | **Format** | **description** |
| --- | --- | --- |
| **Encode** | <IT\_RESOURCE\_KEY>~<ID> | IT\_RESOURCE\_KEY is the numeric code assigned to each IT resource in Identity Manager.  ID is the target system code assigned to each entry in a value list. This value is populated based on the target system attribute name specified in the Encode attribute of the background process for syncing LOVs. |
| **decode** | <IT\_RESOURCE\_NAME>~<VALUE> | IT\_RESOURCE\_NAME is the name of the IT resource in Identity Manager.  VALUE is the target system code assigned to each entry in a value list. This value is populated based on the target system attribute name specified in the Decode attribute of the scheduled lookup field synchronization job. |

The table below shows example entries in the value list Lookup.NextCloud.Groups:

| **Encode** | **decoded** |
| --- | --- |
| **81~admin** | NextCloud VM~admin |

## Deploying the Connector

You must install the connector in OIM. If required, it can also be deployed to a connector connector server, however this document will specifically describe how to deploy it to OIM.

### Deploy the ICF Bundle

The ICF Bundle contains all the Java code which performs all the connector operations to interact with the target system.

The Upload JAR Utlity can be used for this purpose. Refer to the following link at the *Upload JAR Utility* section:

<https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.3/omdev/deploying-and-undeploying-customizations.html#GUID-08C8ABF1-B783-4226-9894-361E26387842>

First copy the following file to any one of the OIM host machines:

* bka-nextcloud-icfbundle.jar

Then execute the UploadJars.sh script.

* When executing the script, be sure to select option **4. ICFBundle**.
* Type the *absolute* path of the bundle file: bka-nextcloud-icfbundle.jar

### Deploy OIM Shared Metadata

#### Import XML Files

1. Log into the OIM sysadmin console.
2. Click the **Import** link.
3. For Files to be imported:
   1. Click **Browse**, and choose **nextcloud-itresource.xml**.
   2. For Import Options screen, click **Next**.
   3. For Summary screen, click **Import**.
   4. Confirm that import was successful by witnessing the following visual confirmation, as well as the absence of any errors:



* 1. Repeat step 3 for all of the following files in sequence:
     1. nextcloud-itresource.xml
     2. nextcloud-lookup.xml
     3. nextcloud-form.xml
     4. nextcloud-form.xml *(Again for a second time because for some reason the prepop adapters fail to map correctly)*
     5. nextcloud-resource.xml
     6. nextcloud- resource.xml *(Again for a second time because for some reason the prepop adapters fail to map correctly)*

#### Configure UI Form

1. You must create and activate a sandbox to use the UI customization and forms management features. You can then publish the sandbox to make the customizations available to other users.

See [Creating a Sandbox](http://www.oracle.com/pls/topic/lookup?ctx=E22999-01&id=OMDEV4798) and [Enabling and Disabling a Sandbox](http://www.oracle.com/pls/topic/lookup?ctx=E22999-01&id=OMDEV4799) in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* .

1. Create a UI Form under the sandbox just created.

[Creating Forms Using the Form Designer](http://www.oracle.com/pls/topic/lookup?ctx=E22999-01&id=OMADM4796) in *Oracle Fusion Middleware Administering Oracle Identity Manager* for instructions on creating a new UI form . When creating the UI form, make sure that you select the resource object that corresponds to the connector you want to associate the form with. Also, check the **Generate Permission Forms box** .

1. Publish the sandbox.

#### Importing the SSL certificate

It may be necessary to import the SSL certificate (if OIM doesn’t already trust the Root CA or intermediate CA). Therefore it may be necessary to execute the steps found in Appendic section: *Importing and Trusted the SSL Certificate*.

### Configure OIM Instance Metadata

This section will define your instance of the target system. For example you may want to define multiple instances for different remote target systems.

#### Preparation

Decide on how you want to name this "instance". This document will refer to this name as [INST\_NAME].

| D:\tt\icon-note-16.png | **Note** |
| --- | --- |
| [INST\_NAME] should be something which specifically differentiates that instance from other possible ones. For example "Nextcloud Local VM" |

#### Configure the IT Resource

The IT resource for the target system contains connection information about the target system.

1. Log into the OIM sysadmin console.
2. Navigate to **IT Resource**.
3. Click **Create IT Resource**.
   1. **IT Resource Name** = [INST\_NAME]
   2. **IT Resource Type** = NextCloud Server
4. **Continue**.
5. Specify IT Resource Parameter Values, as per the following table

| **Parameter** | **Description** |
| --- | --- |
| **Configuration Lookup** | This is the main Lookup entry point for all connector Lookup configuration. Typically you will leave as default unless you are attempting to expanding development to support multiple instances with differing requirements. *See Multiple Targets Support.*  Default: Lookup.NextCloud.Configuration |
| **host** | The hostname of the Nextcloud server.  *Note: this must match the Common Name, or one of the Subject Alternative Names in the certificate served by the same address.*  Example: nextcloud.example.com |
| **port** | Example: 443 |
| **sslEnabled** | true if Https, otherwise false  Example: true |
| **username** | The user account name within Nextcloud which will be used to connect to the target system, and perform all connector operations.  Example: admin |
| **password** | The user account password within Nextcloud which will be used to connect to the target system, and perform all connector operations. |

1. Continue.
2. Continue.
3. Continue.
4. Continue.
5. Finish.

#### Configure the Application Instance

The Application Instance is the entity which is requested in the OIM catalog and associated with users in OIM.

1. Log into the OIM sysadmin console.
2. Navigate to **Application Instances**.
3. Click **Create**.



1. Fill in the following attributes:

| **Attribute** | **Description** |
| --- | --- |
| **Name** | [INST\_NAME] but without any whitespaces (*See Preparation*). |
| **Display Name** | [INST\_NAME] – (*See Preparation*). |
| **Resource Object** | NextCloud |
| **IT Resource** | This is the IT Resource you created in section *Configure the IT Resource*. |
| **Form** | This is the UI Form you created in section *Configure UI Form*. |

1. Click **Save**.

#### Create Schedule Jobs

1. Prepare the schedule job XML file called: **nextcloud-scheduler.xml**
   1. **Search and replace** the following text within the file:

| **Search** | **Replace** |
| --- | --- |
| **[IT\_RESOURCE\_NAME]** | [INST\_NAME] – (*See Preparation*). |
| **[APP\_NAME]** | [INST\_NAME] but without any whitespaces (*See Preparation*). |

1. Log into the OIM sysadmin console.
2. Click the **Import** link.
3. Click **Browse**, and choose **nextcloud-scheduler.xml**.
4. For Import Options screen, click **Next**.
5. For Summary screen, click **Import**.
6. Confirm that import was successful by witnessing the following visual confirmation, as well as the absence of any errors:



## Using the Connector

### Reconciliation Processes

All reconciliation processes are performed using scheduled jobs which is found within the OIM Sysadmin console -> **Scheduler** link.

#### Group Definition Reconciliation

This schedule job is called **[INST\_NAME] Group Lookup Reconciliation**.

Ensure the parameters are as follows before executing:

| **Parameter** | **Value** |
| --- | --- |
| **Code Key Attribute** | \_\_UID\_\_ |
| **Decode Attribute** | \_\_UID\_\_ |
| **Filter** | This is an ICF Filter. See Appendix section *Using ICF Filters*.  Note: For group definition reconciliation, this field is not supported at it’s current development.  *Leave this field empty.* |
| **IT Resource Name** | [INST\_NAME] – (*See Preparation*). |
| **Lookup Name** | Lookup.NextCloud.Groups |
| **Object Type** | Group |

When executed, this schedule job will synchronise all group definitions which are found in the target. This includes deleting records which no longer exist in the target, as well us create + update groups which are still valid.

Additionally, this job will update the ent\_list table in the database, as well as perform catalog synchronization. This means all groups will immediately become available as entitlements as well as available in the OIM Catalog

#### User Create Reconciliation

This job will perform full reconciliation of all account data which exists in the target system (i.e. newly created, or updated).

Ensure the parameters are as follows before executing:

| **Parameter** | **Value** |
| --- | --- |
| **Application Name** | [INST\_NAME] but without any whitespaces (*See Preparation*). |
| **Object Type** | User |

#### User Delete Reconciliation

Warning: Always ensure you execute the *User Create Reconciliation* job prior to executing this job. Failure to do so may result in the unintended deletion of accounts which are still active.

This job will perform full reconciliation of all account data which no longer exists in the target system (i.e. deleted). It works by first identifying which accounts do exist, (essentially using the exact same API calls as the *User Create Reconciliation* job) and then comparing that information with the existing Active accounts in OIM. It does this using pre-existing Reconciliation data, therefore, one must always first execute the *User Create Reconciliation* job first.

Ensure the parameters are as follows before executing:

| **Parameter** | **Value** |
| --- | --- |
| **Application Name** | [INST\_NAME] but without any whitespaces (*See Preparation*). |
| **Object Type** | User |

#### User Search Create Reconciliation

This job will perform full reconciliation of all account data which exists in the target system (i.e. newly created, or updated).

Ensure the parameters are as follows before executing:

| **Parameter** | **Value** |
| --- | --- |
| **Application Name** | [INST\_NAME] but without any whitespaces (*See Preparation*). |
| **Filter** | This is an ICF Filter. See Appendix section *Using ICF Filters* for more information.  The following ICF filters are supported as remote queries and are therefore the recommended use case:   * equalTo('userid', xxxxxx) * contains('userid', xxxxxx) * contains('email', xxxxxx) * contains('displayname', xxxxxx)   All other ICF filters will be executed locally. |
| **Object Type** | User |

### 

### Provisioning Operations

#### Accounts

You provision or request accounts on the Accounts tab of the User Details page within OIM. You can also click the Modify button to change account details.

#### Entitlements

All groups in the target are flagged as Entitilements within OIM. To provision or request an entitlement, navigate to Entitlements tab of the User Details page within OIM.

## Extending the functionality of the Connector

### Reconciliation Operations

#### reconPageSize

This variable is found within the Lookup:

* Lookup.NextCloud.Configuration

This determines how many records will be returned by API http request for the following Schedule Jobs:

* Group Definition Reconciliation
* User Create Reconciliation
* User Delete Reconciliation
* User Search Create Reconciliation - *Note: (for remote requests only).*

Changing this variable may offer improved (or worse) performance in these Schedule Jobs.

### Provisioning Operations

#### postUpdateValidateFields

This variable is found within the Lookup:

* Lookup.NextCloud.Configuration

This determines which form fields will be validated after a provisioning operation. As default it only validates the "phone" field, however it can be updated as a comma separated value (CSV) list. For example: "phone,language"

For any form field in the list, it will perform 2 network requests for every outbound provisioning update operation against that field.

*See section Validating Phone Update Operations for more information.*

## Appendix

### Using ICF Filters

ICF Filters can be classified under 2 different categories depending on how much support is provided by the connector bundle.

* Local queries
* Remote queries

To determine which queries are supported as Remote, refer to the section *Using the Connector*.

#### Local Queries

Local queries are executed by first performing a full reconciliation against the target system. This is very expensive and likely to result in many network requests, and therefore is not recommended.

#### Remote Queries

Remote queries are executed with only 1 network request against the target system. This is therefore very cheap and recommended.

#### Constructing ICF Filters

For official documentation about ICF Filters, and how to construct them, see here:

<https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.3/omdev/integrating-icf-oracle-identity-manager.html#GUID-48E14CF7-D158-4268-9E80-27675BE1DCB4>

The form fields which can be used as "attributeName" can be found in the decoded column of the table within section *Lookup.NextCloud.UM.ReconAttrMap*.

### Importing and Trusted the SSL Certificate

Sometimes we need to configure OIM to trust the Target Application when using SSL to communite with the Target. This is typically experienced when neither the Root certificate nor intermediate certificate is already trusted as an authority for the target.

1. Obtain the SSL public key certificate for the target system.

EITHER:

* 1. Obtain and copy the target system's public key certificate to the computer hosting Identity Manager.

OR

* 1. Execute the following command: *(replace [TARGET\_HOST] and [TARGET\_PORT])*

Openssl s\_client -connect [TARGET\_HOST]:[TARGET\_PORT] -showcerts </dev/null 2>/dev/null|openssl x509 -outform PEM > cert.pem

1. Run the keytool command to import the certificate into the Oracle WebLogic Server keystore:

|  | **command** |
| --- | --- |
| keytool -import -keystore KEYSTORE\_NAME -storepass PASSWORD -file CERT\_FILE\_NAME -alias |

In this command

| **Placeholder** | **Value** |
| --- | --- |
| **KEYSTORE\_NAME** | The full path, including the name, to the WebLogic Server's DemoTrust keystore. |
| **PASSWORD** | The password for the keystore |
| **CERT\_FILE\_NAME** | The full path, including the name, to the target system's certificate. |
| **ALIAS** | The alias for the target system's certificate. |

| D:\tt\icon-note-16.png | **Note** |
| --- | --- |
| SSL can also fail if the OIM system date is not synchronized with the validity date of the certificate. It may be worth double checking that the operating system dates are accurate. |

### Enabling Logging

Execute the steps in this section to enable logging on the connector. Refer to section *Understanding Logging Levels* for more information.

| **Note** |
| --- |
| Sometimes an error is experienced in the form of an uncaught exception. In many cases you won’t need to enable logging in order to see this because it will be caught in the primary server.out log file. Therefore it’s worth checking there for any errors in the first instance.  e.g. oim\_server1*.out* |

1. Open the following file for editing:
   1. [ASERVER\_DOMAIN\_HOME]/config/fmwconfig/servers/*OIM\_SERVER*/logging.xml

| D:\tt\icon-note-16.png | **Note** |
| --- | --- |
| Replace *OIM\_SERVER* with the specific node in the cluster against which you are making this change. Also note that this change is required to be repeated on all nodes across a clustered installation if you want the logging to be active on all nodes. |

1. Add or use an existing log\_handler to output to a file. For example:

|  | **Example** |
| --- | --- |
| <log\_handlers>  <log\_handler name='bka-custom' class='oracle.core.ojdl.logging.ODLHandlerFactory' level='TRACE:32'>  <property name='path' value='${domain.home}/servers/${weblogic.Name}/logs/bka-custom.log'/>  <property name='maxFileSize' value='5242880'/>  <property name='maxLogSize' value='52428800'/>  <property name='encoding' value='UTF-8'/>  </log\_handler> |

1. Add loggers to pick up specific java packages in use during connector operations. For example:

|  | | **Example** |
| --- | --- | --- |
| <logger name='bka.iam.identity' level='TRACE:32'>  <handler name='bka-custom'/>  </logger>  <logger name='oracle.iam.connectors.icfcommon' level='TRACE:32'>  <handler name='bka-custom'/>  </logger> |
| D:\tt\icon-note-16.png | **Note** | |
| bka.iam.identity is a package frequently used by custom connector bundles developed for bka.  oracle.iam.connectors.icfcommon is an OIM propetiary package frequently used by ICF connector operations, and so is useful to gather logging on. | |

1. Save and exit the file.
2. Restart OIM.

### Understanding Logging Levels

Identity Manager verwendet den Protokollierungsdienst Oracle Diagnostic Logging (ODL) zum Aufzeichnen aller Arten von Ereignissen, die den Connector betreffen.

#### Grundlegendes zu Ebenen der Protokollierung

Wenn Sie die Protokollierung aktivieren, speichert Identity Manager automatisch Informationen zu Ereignissen in einer Protokolldatei, die während der Bereitstellungs- und Abstimmungsvorgänge auftreten.

Identity Manager verwendet Oracle Java Diagnostic Logging (OJDL) für die Protokollierung. OJDL basiert auf java.util.logger. Um den Ereignistyp anzugeben, für den die Protokollierung stattfinden soll, können Sie die Protokolle auf eine der folgenden verfügbaren Ebenen festlegen:

| **Ebene** | **Bedeutung** |
| --- | --- |
| **SEVERE.intValue()+100** | Diese Ebene ermöglicht die Protokollierung von Informationen über schwerwiegende Fehler. |
| **SEVERE** | Diese Ebene ermöglicht die Protokollierung von Informationen zu Fehlern, die es Identity Manager ermöglichen könnten, weiter ausgeführt zu werden. |
| **WARNING** | Diese Ebene ermöglicht die Protokollierung von Informationen über potenziell schädliche Situationen. |
| **INFO** | Diese Ebene ermöglicht die Protokollierung von Nachrichten, die den Fortschritt der Anwendung hervorheben. |
| **CONFIG** | Diese Ebene ermöglicht die Protokollierung von Informationen zu detaillierten Ereignissen, die für das Debuggen hilfreich sind. |
| **FINE, FINER, FINEST** | Diese Ebenen ermöglichen die Protokollierung von Informationen zu detaillierten Ereignissen, wobei FINEST Informationen zu allen Ereignissen protokolliert. |

Diese Protokollebenen werden den Kombinationen aus ODL-Nachrichtentyp und -Ebene zugeordnet:

| **Ebene** | **Bedeutung** |
| --- | --- |
| **SEVERE.intValue()+100** | INCIDENT\_ERROR:1. |
| **SEVERE** | ERROR:1 |
| **WARNING** | WARNING:1 |
| **INFO** | NOTIFICATION:1 |
| **CONFIG** | NOTIFICATION:16. |
| **FINE** | TRACE:1. |
| **FINER** | TRACE:16 |
| **FINEST** | TRACE:32 |

Die Konfigurationsdatei für OJDL lautet logging.xml und befindet sich unter folgendem Pfad:  
  
 *DOMAIN\_HOME*/config/fmwconfig/servers/*OIM\_SERVER*/logging.xml

Hier sind *DOMAIN\_HOME* und *OIM\_SERVER* der Domänenname und der Servername, die während der Installation von Identity Manager angegeben wurden.

### Example Raw API User Data Response

{

    "ocs": {

        "meta": {

            "status": "ok",

            "statuscode": 100,

            "message": "OK",

            "totalitems": "",

            "itemsperpage": ""

        },

        "data": {

            "enabled": **true**,

            "storageLocation": "/mnt/ncdata/JOEBLOGGS",

            "id": "JOEBLOGGS",

            "lastLogin": 0,

            "backend": "Database",

            "subadmin": [],

            "quota": {

                "quota": "none",

                "used": 0

            },

            "email": "joe.bloggs@example.com",

            "additional\_mail": [],

            "displayname": "Joe Bloggs",

            "phone": "",

            "address": "",

            "website": "",

            "twitter": "",

            "groups": ["admin"],

            "language": "de",

            "locale": "de\_DE",

            "notify\_email": **null**,

            "backendCapabilities": {

                "setDisplayName": **true**,

                "setPassword": **true**

            }

        }

    }

}