

Upgrade from 1.0.X to 1.1.X

1. Relevant changes
 1. Role provisioning
 2. Changes in the REST interface
 3. MD5 not supported any more
 4. Quartz upgraded to 2.1.X
 5. New mandatory conf param 'notificationjob.cronExpression'
 6. Changes in Task table
 7. Changes in SyncopeRole table
 8. Changes in ConnInstance table
 9. Changes in ExternalResource table
 10. Changes in Entitlement table
2. Suggested procedure
 1. Make the TODO tab empty
 2. Backup
 3. The new 1.1.X project
 4. Build, deploy and run
 5. Perform custom SQL upgrade
 6. Enable ContentUpgrader
 7. Enable SyncopeMD5FallbackAuthenticationProvider
 8. Build, deploy and run
 9. Disable ContentUpgrader
 10. Build, deploy and run

Relevant changes

Role provisioning

Role provisioning (provided by [SYNCOPE-172](#) and related) changed the way how the `schema mapping` is stored by removing the `SchemaMapping` table and introducing the following new tables:

- `UMapping`
- `UMappingItem`
- `RMapping`
- `RMappingItem`

Changes in the REST interface

More details in the [dedicated page](#).

MD5 not supported any more

[SYNCOPE-51](#) removed MD5 from supported password cipher algorithm.

This means that:

1. `password.cipher.algorithm` must be set to something different (say `SHA1`) in `SyncopeConf` table
2. any row in `SyncopeUser` table with `cipherAlgorithm=MD5` must be set to something different (say `SHA1`); moreover, all users must change their own password

Quartz upgraded to 2.1.X

Syncope now features the latest version of Quartz (as per [SYNCOPE-93](#)).

Remove any `QRTZ_*` table from the database: such tables will be re-created at next startup.

New mandatory conf param 'notificationjob.cronExpression'

As per [SYNCOPE-216](#), you can specify when the `NotificationJob` will run.

Add a row to `SyncopeConf` table with `key = notificationjob.cronExpression` and your chosen cron expression as value.

Reference and some examples of cron expressions are provided in [Quartz website](#).

Changes in Task table

- **SYNCOPE-144** introduced two additional columns in Task table:

```
name VARCHAR(255),
description VARCHAR(255)
```

- Change any value from `org.apache.syncope.core.scheduling.SyncJob` to `org.apache.syncope.core.sync.SyncJob` of the `jobClassName` column

Moreover, **SYNCOPE-186** renamed `jobActionsClassName` to `actionsClassName`

Changes in SyncopeRole table

SYNCOPE-225 introduced three additional columns in SyncopeRole table:

```
inheritOwner INTEGER,
ROLEOWNER_ID BIGINT,
USEROWNER_ID BIGINT
```

Changes in ConnInstance table

SYNCOPE-279 introduced an additional column in ConnInstance table:

```
connRequestTimeout INTEGER
```

Changes in ExternalResource table

- `ExternalResource.forceMandatoryConstraint` renamed as `ExternalResource.enforceMandatoryCondition`
- `ExternalResource` has now an additional column

```
actionsClassName VARCHAR(255)
```

Changes in Entitlement table

- **SYNCOPE-319** introduced the new `CONNECTOR_RELOAD` entitlement
- **SYNCOPE-331** renamed `RESOURCE_GETOBJECT` as `RESOURCE_GETCONNECTOROBJECT`

Suggested procedure

Make the TODO tab empty

1. Approve or reject any pending [approval](#)
2. Manage any pending [user request](#)

Backup

Make full backup of

1. the [internal storage](#)
2. your 1.0.X project's (the one [created from archetype](#)) sources
3. configured [bundles and logs](#) directories

The new 1.1.X project

1. [create a new 1.1.X project](#)
2. copy any existing Java class you have developed for the former 1.0.X project and make necessary adaptations
3. include any connector bundle you might [have added](#) in the former 1.0.X project
4. configure the [internal storage](#) as done for the former 1.0.X project

Build, deploy and run

This first run will barely upgrade the existing SQL schema in the internal storage: you will notice many exceptions in the logs files, just ignore.

Once the core application has started (check this from the log files), stop the JEE container.

Perform custom SQL upgrade

At this point you need to manually perform some SQL changes according to the description above.

An example of such changes is reported in the attached [sample for MySQL](#): the set of SQL statements you need to run vary depending on the DBMS used for internal storage.

Enable ContentUpgrader

In `core/src/main/resources/syncopeContext.xml` add

```
<bean class="org.apache.syncope.core.init.SpringContextInitializer">
  <property name="upgrade" value="true"/>
</bean>
```

Enable SyncopeMD5FallbackAuthenticationProvider

If some of existing users have MD5 passwords and you want them being able to log in without prior changing their password, modify in `core/src/main/resources/securityContext.xml`

```
<bean id="syncopeAuthenticationProvider"
      class="org.apache.syncope.core.security.
SyncopeAuthenticationProvider">
  <property name="adminUser" value="\${adminUser}"/>
  <property name="adminPassword" value="\${adminPassword}"/>
  <property name="adminPasswordAlgorithm"
value="\${adminPasswordAlgorithm}"/>
  <property name="syncopeUserDetailsService" ref="
syncopeUserDetailsService"/>
</bean>
```

to

```
<bean id="syncopeAuthenticationProvider"
      class="org.apache.syncope.core.security.
SyncopeMD5FallbackAuthenticationProvider">
  <property name="adminUser" value="\${adminUser}"/>
  <property name="adminPassword" value="\${adminPassword}"/>
  <property name="adminPasswordAlgorithm"
value="\${adminPasswordAlgorithm}"/>
  <property name="syncopeUserDetailsService" ref="
syncopeUserDetailsService"/>
</bean>
```

Once all of your users have changed their password, you can restore the original configuration.

`SyncopeMD5FallbackAuthenticationProvider` can also be taken as reference to implement [more sophisticated handling](#) of existing MD5 passwords.

Build, deploy and run

This second run should succeed without logging any exception.

You should now have full access to all Syncope features.

Stop the JEE container

Disable ContentUpgrader

In `core/src/main/resources/syncopeContext.xml` remove

```
<bean class="org.apache.syncope.core.init.SpringContextInitializer">
  <property name="upgrade" value="false"/>
</bean>
```

Build, deploy and run

This final run can be considered as definitive: only, if you configured `SyncopemD5FallbackAuthenticationProvider` or similar, consider planning its removal in the near future, maybe once all users have changed their password.