

# Synchronization - time slices of contractual relationship

sync, contract, slice



**The synchronization of time slices of identity's (contractual) relationship** works according to the same rules as identity synchronization. In this page we will describe only behavior specific for this synchronization.



**Synchronization of contract slice** uses a configuration slightly different from the synchronization of contract. **Valid till date** is set only for the last slice that's synchronized!

## What is the time slice of contractual relationship

Typically one contractual relationship is equal to one contract in company for the identity. One slice of the contractual relationship describes what the contract looked like in a specific time slot, i. e., it is a snapshot of the contract at a given time. Sync of slices persists slices to the entity `IdmContractSlice` and creates relation to the account with entity `AccContractSliceAccount`. If more slices exist for the same contract, then only one is being used as source at one time (the latest). See [here](#) to learn more about slices.

From these slices (specifically the latest one), a contract is built. This contract is managed only by time slices and cannot be modified directly.

## Actions after end of sync

After the synchronization, HR processes must be run. In the Specific settings of synchronization, you can check **After end, start the HR processes** and not have to worry about anything. This, however, has its problems. If you sync a large amount of time slices (>10,000), you can see some issues. In those cases, it is recommended you run the HR processes manually.



If you manually cancel the synchronization, the HR processes **will not start** even if you checked After end, start the HR processes. You must run them manually.

## Running HR processes manually

As mentioned above, sometimes, you need to run HR processes manually. Each HR process is a long-running task and can be run from Settings > Task scheduler > Scheduled task. There you can run the

desired LRT by simply clicking the green arrow button.

You should run HR processes in this order:

1. ``ClearDirtyStateForContractSliceTaskExecutor``
2. ``SelectCurrentContractSliceTaskExecutor``
3. ``HrEnableContractProcess``
4. ``HrEndContractProcess``
5. ``HrContractExclusionProcess``

For more details about some of them, see below.

You always have to run at least the ``ClearDirtyStateForContractSliceTaskExecutor`` LRT, even if you cancelled the synchronization manually. If you don't, you will see duplicated contract as a result. This HR process creates and deletes contracts and deletes slices which are supposed to be deleted.



So, in a situation where you ran the synchronization and either cancelled it manually before it finished and the HR processes ran or just decided to sync your contracts again (perhaps the data changed), **before you run the synchronization a second time, you must run the ``ClearDirtyStateForContractSliceTaskExecutor``**. You can ignore the rest of HR processes for now but you absolutely must run ``ClearDirtyStateForContractSliceTaskExecutor``.

## Clear dirty state for contract slice

This LRT deletes those slices which have the flag "dirty" which means that they should be deleted (they are obsolete). It also creates and deletes contracts from the time slices. Beware that it takes some time to complete.

## Select current contract slice

Before HR processes proper are executed, this long running task selects the current contract slices. The LRT is started synchronously and complete log will contain information about its run.

## HR processes proper



**In new IdM installation must HR processes be running at least once manually or by trigger.** Without this, sync will be not able **find HR tasks** (warning will be logged in sync log)!

[HR processes](#), simply put, ensure the correct state of identity depending on the state of their

contractual relationships. Because we need to evaluate the status of contractual relationships as a whole (to a given identity), it is not possible to trigger HR processes during the synchronization of each contractual relationship. Therefore, no HR processes are executed during synchronization but only after it ends.

There are three HR processes: HrEnableContractProcess, HrEndContractProcess, and HrContractExclusionProcess. Their names are descriptive enough but in a nutshell HrEnableContractProcess determines that the contract should start (it's valid), HrEndContractProcess determines that the contract should end (it's no longer valid), and HrContractExclusionProcess deals with contracts exclusion's start and end. (e. g., due to parental leave)

HR processes can be (**should be**) correctly started after the end of the sync. This can be ensured ticking the checkbox `After end, start the HR processes` in the Specific settings of synchronization in sync configuration. If is this property ticked, then all HR processes including ClearDirtyStateForContractSliceTaskExecutor and SelectCurrentContractSliceTaskExecutor will be run once the synchronization finishes.

## Automatic roles

Recalculation of automatic roles is skipped during sync but it can be (**should be**) started after the end of the sync. This can be ensured by the property '**After end, start the automatic role recalculation**' in the Specific settings of synchronization of sync configuration.

### Synchronization details

SettingsSpecific settingsFilterLogs

**Default type of structure**

Organization structure (ORGANIZATIONS)

Relations will be assigned (searched) for positions within the selected structure.

**Default position in structure**

Select or type to search ...

If the position for a given relation is not filled in the source system, then this will be used as the default.

**Default leader**

Select or type to search ...

If the none leader for a given relation is filled in the source system, then this will be used as the default.

☐ After end, start the HR processes

After successful synchronization, HR processes will be executed.

☐ After end, start the automatic role recalculation

After successful synchronization will be start recalculation of all automatic role by attribute.

## Fields for sync contractual relationship mapping

- **Contract code** - Code of the parent contract. This `String` value represents relation between all slices for the same contract.
- **Valid from of slice** - Defines time from which the slice is valid. Valid till of slice is computed automatically (by the validity of the next slice) after save. Output value from attribute transformation must be 'org.joda.time.LocalDate' (or String in the format `yyyy-MM-dd`).
- **Owner** - Relation's owner, it must be an existing identity in IdM. This field is required for every relation. Output from attribute transformation can be:
  - ID of IdM identity in String or UUID format.
  - Username of IdM identity in String.
- **Main** - Defines if the contract is the main contract (among all contracts for the identity). Output from attribute transformation must be Boolean.
- **State** - State of contract. Output from attribute transformation must be enumeration ContractState or String representation for this enumeration (DISABLED, EXCLUDED) (see below for more details).
- **Position** - String representation of contract. Typically name of contract.
- **Guarantees** - List of leaders, directly linked on the contractual relationship (see below for more details).
- **Work position** - Defines link to some tree node. Generally defines a place in the organization structure (see below for more details).
- **Valid from of contract** - Validity for the contractual relationship. Output value from attribute transformation must be 'org.joda.time.LocalDate' (or String in the format `yyyy-MM-dd`).
- **Valid till of contract** - Validity for the contractual relationship. Output value from attribute transformation must be 'org.joda.time.LocalDate' (or String in the format `yyyy-MM-dd`).
- **External contractor** - If the contractual relationship is for external identity, then the output value is boolean true.
- **Description** - String for the description of the relation.

Link to SQL details for making the Views:

[https://wiki.czechidm.com/priv/time\\_slices#mssql\\_update\\_of\\_sql\\_query\\_and\\_extra\\_information](https://wiki.czechidm.com/priv/time_slices#mssql_update_of_sql_query_and_extra_information)

### Guarantees field

List of leaders, directly linked to the contractual relation. Linked leader must exists in IdM. Output from attribute transformation can be:

- Username of leader (String).
- Id of leader (UUID or String).
- List of usernames (List<String>).
- List of Ids (List<String> or List<UUID>).
- Null value. If is value not defined and in sync configuration has set '**Default leader**', then will be this leader set to relation.

If no leader is found, the synchronization item will be marked as 'warning' (relation will be created/saved). Detailed information will be saved in item log:

.....

```
Finding guarantee [temslie7].  
.....  
Warning! - Identity [temslie7] was not found for [temslie7]!  
.....
```

## Work position field

Defines link to some tree node. Generally defines a place in the organization structure. Output from attribute transformation can be:

- Id of a tree node (UUID or String).
- Code of a tree node. Node by code will be looked up in default tree (defined in sync configuration '**Default type of structure**').
- Null value. If no value is defined and the sync configuration has set '**Default position in structure**', then this node will be set as the tree node.

If no node is found the synchronization item will be marked as 'warning' (relation will be created/saved). Detailed information will be saved in item log:

```
.....  
Work position - try find directly by transformed value [Divanoodle]!  
.....  
Work position - was not not found directly from transformed value  
[Divanoodle]!  
.....  
Work position - try find in default tree type [DEFAULT_ORG] with code  
[Divanoodle]!  
.....  
Warning - Work position - none node found for code [Divanoodle]!
```



If no work-position attribute is defined in the mapping, then **no** default position will be set.

## State field

State of contract. Output from attribute transformation must be enumeration ContractState or String representation for this enumeration.

ContractState has these values:

- **DISABLED**
- **EXCLUDED**

From:  
<https://wiki.czechidm.com/> - **CzechIdM Identity Manager**

Permanent link:  
<https://wiki.czechidm.com/devel/documentation/synchronization/dev/contract-slice-sync>

Last update: **2020/04/06 18:25**

