

# Systems - DB: TreeNodes (org. structure) provisioning

## What do you need before you start

- You need install **CzechIdM 7.7.0** (and higher).
- You need be logged in as **admin**.
- You need enable **ACC** module.
- You must have database **jdbcTest** with table **organizations**:

```
CREATE DATABASE "jdbcTest"
WITH OWNER = idmadmin
    ENCODING = 'UTF8'
    TABLESPACE = pg_default
    LC_COLLATE = 'cs_CZ.UTF-8'
    LC_CTYPE = 'cs_CZ.UTF-8'
    CONNECTION LIMIT = -1;
```

```
CREATE TABLE public.organizations
(
    id character varying(255),
    code character varying(255),
    name character varying(255),
    parent character varying(255)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.organizations
    OWNER TO idmadmin;
```

## Create and edit DB system

We have completely empty organization's agenda.

czechidm

Development



admin





Profile

Tasks

Users

Organization

Roles

Systems

Audit

Notifications

Settings

Organizational structure

StructureUsersStructure elements

+ Add

Filter

Cancel filter

Filter

Code / Name

Superior element

Recursively by tree structure down

No results found

0 - 0 of 0 records

BCV solutions s.r.o. | Help | ServiceDesk | About app

Now we go into **Systems** agenda and add new system. Click on green button **Add**.

czechidm

Development



admin





Profile

Tasks

Users

Organization

Roles

Systems

Audit

Systems

+ Add

Filter

No results found

0 - 0 of 0 records

BCV solutions s.r.o. | Help | ServiceDesk | About app

Fill **System name** and at bottom of page there is green button **Save and continue**, so click on it.

New system

System name

test - organization

☐ Use remote connector server

After check, local connectors will not be used.

Password policy for validation

Password policy for validation

Password policy for generating

Password policy for generating

Description

Go to **Configuration** tab and fill in information about your database. Ours are as such:

|               |                                             |   |
|---------------|---------------------------------------------|---|
| ^ Host        | ^ jdbc:postgresql://localhost:5432/jdbcTest | ^ |
| Port          | 5432                                        |   |
| User          | idmadmin                                    |   |
| User Password | *****                                       |   |
| Database      | jdbcTest                                    |   |

https://wiki.czechidm.com/

Printed on 2024/03/30 03:51

|                     |                                           |  |
|---------------------|-------------------------------------------|--|
| Table               | organizations                             |  |
| Key Column          | id                                        |  |
| JDBC Driver         | org.postgresql.Driver                     |  |
| JDBC Connection URL | jdbc:postgresql://localhost:5432/jdbcTest |  |



Database Table Connector (connId)

Test connector

Connector configuration

Pooling configuration

Additional connector configuration

Name Quoting

Select whether database column names for this resource should be quoted, and the quoting characters. By default, database column names are not quoted (None). For other selections (Single, Double, Back, or Brackets), column names will appear between single quotes, double quotes, back quotes, or brackets in the SQL generated to access the database.

Host

localhost

Enter the name of the host where the database is running.

Port

5432

Enter the TCP port number the database server is listening on.

User

idmadmin

Enter the name of the mandatory Database user with permission to account table.

User Password

•••••

Enter a user account password that has permission to access accounts table.

Database

jdbcTest

Enter the name of the database on the database server that contains the table.

Table

organizations

Enter the name of the table in the database that contains the accounts.

Key Column

id

This mandatory column value will be used as the unique identflier for rows in the table.

Password Column

Enter the name of the column in the table that will hold the password values. If empty, no validation on resource and passwords are activated.

Status Column

Enter the name of the column in the table that will hold the status values. If empty enabled and disabled operation wont be performed.

Disabled Status Value

false

Enter the value for disabled status. Default is "false".

Enabled Status Value

true

Enter the value for enabled status. Default is "true".

Default Status Value

true

Enter the value for status in case of status not specified. Default is "true".

JDBC Driver

org.postgresql.Driver

Specify the JDBC Driver class name. Oracle is `oracle.jdbc.driver.OracleDriver`. MySQL is `org.gjt.mm.mysql.Driver`. PostgreSQL is `org.postgresql.Driver`. Could be empty if datasource is provided.

JDBC Connection URL

jdbc:postgresql://%h:%p/%d

Specify the JDBC Driver Connection URL. Oracle template is `jdbc:oracle:thin:@%h:%p:%d`. MySQL template is `jdbc:mysql://%h:%p/%d`. PostgreSQL template id `jdbc:postgresql://%h:%p/%d`, for more info, read the JDBC driver documentation. Could be empty if datasource is provided.

☐ Enable writing empty string

Select to enable support for writing an empty strings, instead of a NULL value, in character based columns defined as not-null in the table schema. This option does not influence the way strings are written for Oracle based tables. Default is "false".

☒ Rethrow all SQLExceptions

If this is not checked, SQL statements which throw SQLExceptions with a 0 ErrorCode will be have the exception caught and suppressed. Check it to have exceptions with 0 ErrorCodes rethrown. Default is "true".

☐ Native Timestamps

Select to retrieve Timestamp data type of the columns in java.sql.Timestamp format from the database table. Default is "false".

☐ All native

Select to retrieve all data type of the columns in a native format from the database table. Default is "false".

Validate Connection Query

There can be specified the check connection alive query. If empty, default implementation will test it using the switch on/off the autocommit. Some select 1 from dummy table could be more efficient.

Change Log Column (Sync)

The change log column store the latest change time. Providing this value the Sync capabilities are activated.

Datasource Path

Enter the JDBC Data Source Name/Path to connect to the Oracle server. If specified, connector will only try to connect using Datasource and ignore other resource parameters specified. An the example value is: `jdbc/sampledatasourceName`

Initial JNDI Properties (multi)

Could be empty or enter the JDBC JNDI Initial context factory, context provider in a format: key = value.

Password cipher algorithm

CLEARTEXT

Cipher algorithm used to encode password before to store it onto the database table. Specify one of the values among CLEARTEXT,AES, MD5, SHA1, SHA256 or a custom implementation identified by its class name. Default is "CLEARTEXT".

☐ Force password encoding to upper case

Force password encoding to upper case. Default is "false".

Password cipher key

Specify key in case of reversible algorithm.

☐ Force password encoding to lower case

Force password encoding to lower case. Default is "false".

☐ Retrieve password

Specify if password must be retrieved by default. Default is "false".

Resource Password Charset

UTF-8

Specify Character set used by resource to encode clear text password. Default is "UTF-8".

Then go to tab **Scheme** and generate scheme.

The screenshot shows the 'test - organization' system details page. The left sidebar contains a menu with the following items: Basic information, Configuration, Provisioning brake, Accounts, Entities, **Scheme** (highlighted with a red box), Mapping, Synchronization, and Provisioning. The main content area is titled 'System scheme' and features a green 'Generate scheme' button (highlighted with a red box). Below this, the 'Object types in system' section displays a table with the following data:

| Object name                                                        | Auxiliary                | Container                | Id      |
|--------------------------------------------------------------------|--------------------------|--------------------------|---------|
| <input type="checkbox"/> <input type="checkbox"/> <b>_ACCOUNT_</b> | <input type="checkbox"/> | <input type="checkbox"/> | 6885b4c |

At the bottom right of the table, it says '1 - 1 of 1 records'. The footer of the page includes 'BCV solutions s.r.o. | Help | ServiceDesk | About app' and an upward arrow icon.

Then go to **Mapping** and click on **Add** button.

The screenshot shows the 'test - organization' system details page with the 'Mapping' tab selected in the sidebar (highlighted with a red box). The main content area is titled 'Attributes mapping attribute groups for operation and entity type' and features a green '+ Add' button (highlighted with a red box). Below this, a light blue box displays the message 'No results found'. At the bottom right, it says '0 - 0 of 0 records'.

Now fill in information from picture bellow and click on **Save and continue**.

test - organization

System details

Basic Information

Configuration

Provisioning brake

Accounts

Entities

Scheme

Mapping

Synchronization

Provisioning

Mapping of attributes for IdM entity and operation type

Detail

Account management

Operation type

Provisioning

Mapping name

provisioning

Object name

\_\_ACCOUNT\_\_

Entity type

Tree

Tree type

Organization structure (ORGANIZATIONS)

Back

Save and continue

At the bottom of page, there are mapped attributes. We are going to add 4 attributes:

| ^ Attribute in schema attr. | ^ Entity field | ^ Name     | ^ IdM key | ^ Identifier | ^ Entity attr. | ^ Extended |
|-----------------------------|----------------|------------|-----------|--------------|----------------|------------|
| code                        |                | code       |           | false        | true           | false      |
| Code (String)               |                |            |           |              |                |            |
| name                        |                | name       |           | false        | true           | false      |
| Name (String)               |                |            |           |              |                |            |
| parent                      |                | parent     |           | false        | true           | false      |
| Parent (String)             |                |            |           |              |                |            |
| __NAME__                    |                | __NAME__   |           | true         | false          | true       |
|                             |                | externalId |           |              |                |            |

Fill "code, name, parent" as in the picture:

test - organizationSystem details

Basic Information

Configuration

Provisioning brake

Accounts

Entities

Scheme

Mapping

Synchronization

Provisioning

Attribute mapping details

Disabled

Mapping name

provisioning (Tree - Provisioning)

Attribute in schema

code (\_ACCOUNT\_)

Name

code

User-defined name of the attribute

Strategy

Set value as it is

Send always

Send IdM value only if its not null

Identifier

Entity attr.

Extended attr.

Main form definition is supported only.

Confidential attr.

Authentication attr.

Attribute used for authentication on connected system.

Include on password change

Send this attribute into provisioning, when password is changed.

The value is cached

The attribute value will be saved and read from the cache. At this moment, it is used only in sync. The key is this attribute and attributes from the end system (lcAttribute). The value is the transformed value from the end system.

Entity field

Code (String)

IdM key

code

Name of entity attr., name of extended attr., or key to confidential storage.

Transformation from system

1

Insert script

Allows value to be transformed from system into a form suitable for CzechidM. Input parameters of this Groovy script are value of the attribute 'attributeValue' and list 'lcAttributes' of object attributes in system.

Transformation to system

1

Insert script

Allows value to be transformed from CzechidM into a form suitable for connected system. Input parameters of this Groovy script are value of attribute 'attributeValue', IdM entity 'entity' and account identifier 'uid'. If output value is empty, system automatically uses available account identifier (uid).

Back

Save

BCV solutions s.r.o. | Help | ServiceDesk | About app

But "\\\_NAME\\\_" fill like this picture:

https://wiki.czechidm.com/

Printed on 2024/03/30 03:51



test - organization

System details

Basic Information

Configuration

Provisioning brake

Accounts

Entities

Scheme

Mapping

Synchronization

Provisioning

Attribute mapping details

Disabled

Mapping name

provisioning (Tree - Provisioning)

Attribute in schema

\_\_NAME\_\_ (\_\_ACCOUNT\_\_)

Name

\_\_NAME\_\_

User-defined name of the attribute

Strategy

Set value as it is

Send always

Send IdM value only if its not null

Identifier

Entity attr.

Extended attr.

Main form definition is supported only.

Confidential attr.

Authentication attr.

Attribute used for authentication on connected system.

Include on password change

The value is cached

Send this attribute into provisioning, when password is changed.

The attribute value will be saved and read from the cache. At this moment, it is used only in sync. The key is this attribute and attributes from the end system (lcAttribute). The value is the transformed value from the end system.

Entity field

Select ...

IdM key

externalId

Name of entity attr., name of extended attr., or key to confidential storage.

Transformation from system

1

Insert script

Allows value to be transformed from system into a form suitable for CzechIdM. Input parameters of this Groovy script are value of the attribute 'attributeValue' and list 'lcAttributes' of object attributes in system.

Transformation to system

1

import java.util.UUID;

2

3

if (attributeValue == null) {

4

return UUID.randomUUID().toString();

5

}

6

return attributeValue;

Insert script

Allows value to be transformed from CzechIdM into a form suitable for connected system. Input parameters of this Groovy script are value of attribute 'attributeValue', IdM entity 'entity' and account identifier 'uid'. If output value is empty, system automatically uses available account identifier (uid).

Back

Save

Now we have 4 mapped attributes:

| Mapped attributes                                       |                                       |            |                                     |                                     |                                     |                          |                                     |
|---------------------------------------------------------|---------------------------------------|------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|
| <div><div>+ Add</div><div>Filter</div><div></div></div> |                                       |            |                                     |                                     |                                     |                          |                                     |
| <input type="checkbox"/>                                | Name                                  | IdM key    | Identifier                          | Entity attr.                        | Extended attr.                      | Transform from system    | Transform to system                 |
| <input type="checkbox"/>                                | <input type="text" value="code"/>     | code       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            |
| <input type="checkbox"/>                                | <input type="text" value="name"/>     | name       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            |
| <input type="checkbox"/>                                | <input type="text" value="__NAME__"/> | externalId | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <input type="checkbox"/>                                | <input type="text" value="parent"/>   | parent     | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            |
| 1 - 4 of 4 records                                      |                                       |            |                                     |                                     |                                     |                          |                                     |

Go to the section **Organization** (in the left menu) to **Structure elements** tab and **Add** new organization.

Profile

Tasks

Users

**Organization**

Roles

Systems

Audit

Notifications

Settings

Organizational structure

Structure

Users

**Structure elements**

+ Add

Filter

Code / Name

Superior element

Recursively by tree structure down

No results found

0 - 0 of 0 records

BCV solutions s.r.o. | Help | ServiceDesk | About app

Fill in information of root organization:

New element

Structure type

Organization structure (ORGANIZATIONS)

Code

rt

Name

root

Superior element

Select or type to search ...

☐ Inactive

Back

Save and continue

Then add another organization, but this one will be child of root:

New element

Structure type

Organization structure (ORGANIZATIONS)

Code

org1

Name

organization1

Superior element

root (rt)

☐ Inactive

Back

Save and continue

As you can see in section **Organization** we have two structure elements.

czechidm

Development



admin

Profile

Tasks

Users

Organization

Roles

Systems

Audit

Notifications

Settings

Organizational structure

Structure

Users

Structure elements

root (1)

organization1

+ Add

Filter

Code / Name

Superior element

Cancel filter

Filter

Recursively by tree structure down

|                          | Code | Name          | Superior element | Inactive                 | Id      |
|--------------------------|------|---------------|------------------|--------------------------|---------|
| <input type="checkbox"/> | org1 | organization1 | root             | <input type="checkbox"/> | d279f4a |
| <input type="checkbox"/> | rt   | root          |                  | <input type="checkbox"/> | 7a99aa7 |

1 - 2 of 2 records

BCV solutions s.r.o.

Help

ServiceDesk

About app

And in database:

|   | id<br>character varying(255)         | code<br>character varying(255) | name<br>character varying(255) | parent<br>character varying(255)     |
|---|--------------------------------------|--------------------------------|--------------------------------|--------------------------------------|
| 1 | 15fc6b36-a394-4dcf-962c-ed24e0d3c3a9 | rt                             | root                           |                                      |
| 2 | 83f333c0-363c-4ffe-a1ad-6f4f16ad64fd | org1                           | organization1                  | 15fc6b36-a394-4dcf-962c-ed24e0d3c3a9 |

If you have it in a state like in the upper picture you successfully complete this tutorial.



Thank you for trying this tutorial, I hope it was useful.

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

Permanent link:  
[https://wiki.czechidm.com/tutorial/adm/systems\\_db\\_treenodes\\_provisioning](https://wiki.czechidm.com/tutorial/adm/systems_db_treenodes_provisioning)

Last update: 2018/03/05 08:53



