

8MAN

Access Rights Management. **Only much Smarter.**



Access Rights Management Using Templates

Version 9

© 2018 Protected Networks GmbH

1	Structure of the document	4
2	Take advantage of customized templates	5
3	Load templates in 8MAN	6
4	Create templates	7
4.1	Edit and name templates	7
4.2	All templates	8
4.2.1	In the header of the template	8
4.2.2	Input options	9
4.2.2.1	Availability of input types	9
4.2.2.2	Basic structure of an input option	10
4.2.2.2.1	Frequent properties	11
4.2.2.2.2	Constraints	14
4.2.2.2.3	MultiLanguageTemplates	15
4.2.2.2.4	Creation rules	16
4.2.2.2.4.1	LookupTable	19
4.2.2.2.5	Hide input fields	21
4.2.2.3	TextField	22
4.2.2.4	TextArea	23
4.2.2.5	MultiValueText	24
4.2.2.6	DropDownList	25
4.2.2.7	FixedValue	26
4.2.2.8	Checkbox	27
4.2.2.9	Radio-Buttons	28
4.2.2.10	AccountSearchTextField	29
4.2.2.11	SearchField	31
4.3	Templates for users/groups/contacts	32
4.3.1	Customize templates for new users	32
4.3.1.1	Enter Name and OU	32
4.3.1.2	Enter additional LDAP attributes	33
4.3.1.3	Assign group memberships	34
4.3.1.4	Run an external program	35
4.3.1.5	Enter password options	36
4.3.1.6	Activation options	37
4.3.1.7	Create an Exchange mailbox	38
4.3.2	Customize templates for new groups	41
4.3.2.1	Preset group options (group type/scope)	41
4.3.2.2	Preset group members	42

4.3.2.3	Enable e-mail (create distribution group) in Exchange	43
4.3.3	Customize templates for new contacts	45
4.3.4	Make templates for users/groups/contacts available in the Web client	47
4.4	Open Order templates	48
4.4.1	Structure of an Open Order template	49
4.4.2	Create an input option	50
4.4.3	Specific Open Order input options	51
4.4.3.1	AccountSearchTextField	51
4.4.3.2	Radio-Buttons	53
4.4.4	Spezifische Open Order Textanzeigen	0
4.4.5	Spezifische Open Order Sichtbarkeitssteuerungen und Gültigkeitsprüfungen	0
4.4.6	Open Order Vorlagen in den 8MATE GrantMA einbinden	54
4.4.6.1	Enter the template's call into the XML Resource Configuration	55
4.4.6.2	Upload an XML resource configuration to the Data Owner configuration	56
4.4.6.3	Set the Open Order resource to requestable	57

1 Structure of the document

With 8MAN you can use customized templates for:

- creating users
- creating groups
- creating contacts (Rich Client only)
- Open Order

Customize the templates according to your needs to standardize, simplify and accelerate the creation of objects. We describe the advantages of individualized templates in the [following chapter](#).

In the section "All template types", we describe the blocks of the templates: The input options with their [properties](#), [constraints](#) and [CreationRules](#). These are the same for all types of templates.

In the following chapters we deal with the specific elements of the templates for [users](#), [groups](#), [contacts](#) and [Open Order](#).

2 Take advantage of customized templates

8MAN provides a set of standard templates, for example for the creation of new users. Based on the template, 8MAN generates the input masks. Use templates customized to your needs and create new objects in a standardized, simplified and accelerated way. The most important advantages are described below:

Dropdown menus and lookups

Assign input fields with drop-down menus from which users can choose. Depending on the selection, additional fields can be filled automatically. You avoid incorrect entries, accelerate the input and have a standardized result.

Example

For the location field, choose "Berlin", "Munich", "Vienna". Based on the selection, the fields "Street", "ZIP" and "City" are filled automatically.

Validation rules

Validate whether the value entered matches certain rules.

Examples

- Minimum length for a password
- Check the format of a telephone number

Group memberships

When creating a user, specify in which groups the new user becomes a member.

Required fields

Specify which entries must be made (must not remain empty)..

Set default values

Assign fields with default values - changeable or not changeable.

Creation rules

Determine how a resulting field is filled from entries that have already been made.

Example

The login name and the email address are automatically generated from the first name and last name.

Hide input fields

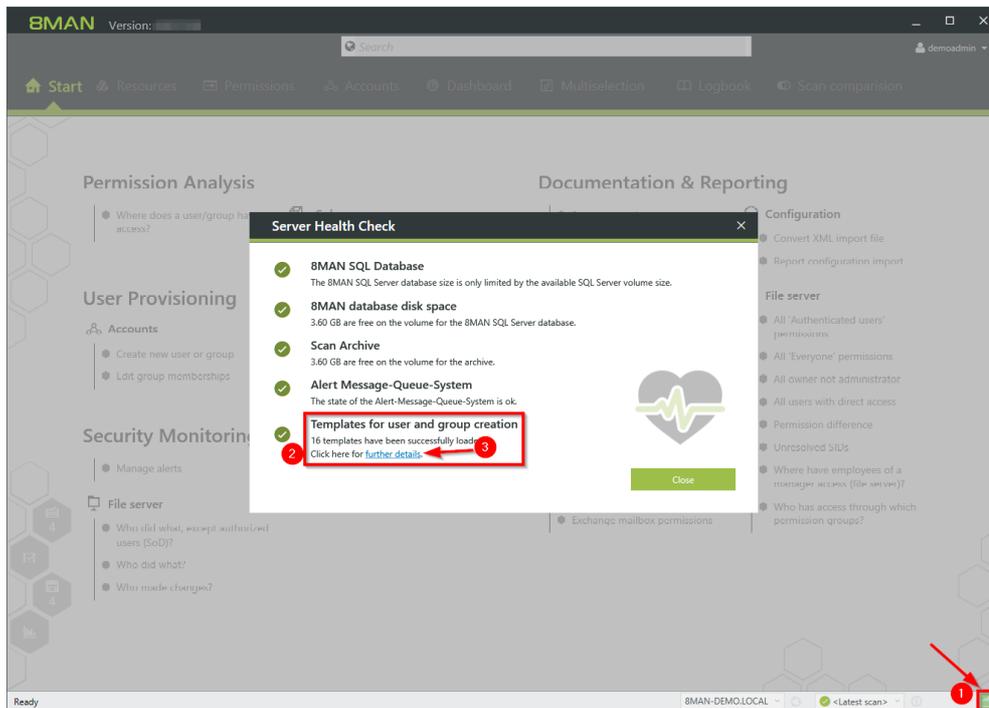
If certain inputs are not required in your company or are already filled with defaults, hide individual input fields or entire areas. This reduces the complexity of the input masks.

3 Load templates in 8MAN

Save Templates in the directory:

`%ProgramData%\protected-networks.com\8MAN\data\templates`

This directory is constantly monitored by 8MAN on new templates (file watcher). New templates will automatically load 8MAN based on the .json file extension. LDAP attributes used in templates are loaded dynamically.



If errors occur when the templates are loaded, they are displayed in the server health check.

You will also find information on the conditions (constraints) applied here.

4 Create templates

4.1 Edit and name templates

8MAN provides sample templates in the directory:

```
%ProgramData%\protected-networks.com\8MAN\data\templates
```

Remove the extension ".example" and assign the desired filename.

The filenames must end as follows depending on the template type:

- `.CreateNewUser.json`
- `.CreateNewGroup.json`
- `.CreateMailContact.json` (*rich client only*)
- `.OpenOrderTemplate.json`

Example

```
"NewUserSales.CreateNewUser.json"
```

The templates use the JSON format. Customize the templates with appropriate editors.

We recommend using [Visual Studio Code](#) (with syntax check) or [notepad++](#).

You can find more information about the JSON format on [Wikipedia](#).

4.2 All templates

4.2.1 In the header of the template

"Version": 1

Leave the value 1.

A value prepared / reserved for future versions of templates.

"TemplateType":

Specify the type of the template. The following types are available:

- CreateNewUser
- CreateNewGroup
- CreateMailContact (*rich client only*)
- OpenOrderRequest

The entry must match the [file name](#).

"Id":

Assign a unique ID. The format is freely selectable. Doubled IDs result in an error message in the [server health check](#).

We recommend using a GUID, e.g. from [guidgen.com](#).

"Displayname":

Assign a name for the template. The name is displayed to the 8MAN user in the template selection (rich client and GrantMA).

"Description":

Assign a description which is also displayed to the user when the template is selected and helps to further distinguish it.

"FullQualifiedDomainName":

Specify the FQDN of the domain. Templates can only be mapped to one domain and are only available in this domain.

The value defined here is available as variable [\[fqdn\]](#) for [creation rules](#).

4.2.2 Input options

4.2.2.1 Availability of input types

	User	Group	Contact	Open Order
<u>TextField</u>	✓	✓	✓	✓
<u>TextArea</u>	✓	✓	✓	✓
<u>MultiValueText</u>	✓	✓	✓	✓
<u>DropDownList</u>	✓	✓	✓	✓
<u>FixedValue</u>	✓	✓	✓	✓
<u>SearchField</u>	✓	✓	✓	✗
<u>Checkbox</u>	✓	✓	✗	✓
<u>Radio</u>	✗	✗	✗	✓
<u>AccountSearchTextField</u>	✗	✗	✗	✓
GroupAccountSearchTextField	✗	✗	✗	✓
GroupAccountSearchTextField	✗	✗	✗	✓
DatePicker	✗	✗	✗	✓
DateRangePicker	✗	✗	✗	✓
PasswordField	✗	✗	✗	✓
Numeric	✗	✗	✗	✓

4.2.2.2 Basic structure of an input option

With an input option, you create the prerequisite for the user to enter data into a form.

Prior to the actual input possibility, the allocation, e.g. for which LDAP attribute the input is to be made.

Example of an assignment in templates for users / groups / contacts

```
"Name": "sn",  
"Definition": {  
  Properties listing  
}
```

The properties define how the input option is displayed and how it behaves.

4.2.2.2.1 Frequent properties

"Type":

Specifies the type of the input field.

Optional: no

Characteristics: **This entry must be the first within the definition.**

Possible values: Depending on the template type. An overview of the available types can be found [here](#).

Default value:

Example: **"Type": TextArea**

"Label":

The label of the input field to be displayed.

Optional: yes

Characteristics: multilanguage is possible

Possible values: any text

Default value:

Example: **"Label": "Comment"**

"DefaultValue":

A value already pre-filled when the form is loaded.

Optional: yes

Characteristics: depends on [Type](#), see [TextArea](#), [MultiValueText](#), [DropDownList](#)

Possible values: any text

Default value:

Example: **"DefaultValue": "This is a pre-filled value."**

"IsEnabled":

Indicates whether the field is editable.

Optional: yes

Characteristics: Fields that can not be edited must not be required fields.

Possible values: **true or false**

Default value: **true**

Example: **"IsEnabled":false**

"IsRequired":

Indicates whether the field is a mandatory field.

<i>Optional:</i>	yes
<i>Characteristics:</i>	Required fields must not be disabled (IsEnabled).
<i>Possible values:</i>	true or false
<i>Default value:</i>	false
<i>Example:</i>	"IsRequired":true

"Description":

Description of the field for display in the tooltip.

<i>Optional:</i>	yes
<i>Characteristics:</i>	multilanguage is possible
<i>Possible values:</i>	any text
<i>Default value:</i>	
<i>Example:</i>	"Description": "Automatically created, non-modifiable comment."

"Items":

An items list for a drop down menu.

<i>Optional:</i>	no
<i>Characteristics:</i>	used only in DropDownList
<i>Possible values:</i>	listing
<i>Default value:</i>	
<i>Example:</i>	<pre>"Items": [{ "Value": "Berlin", "DisplayValue": "Berlin - Germany" }, { "Value": "Paris", "DisplayValue": "Paris - France" }]</pre>

"DisplayValue":

Value displayed in conjunction with [Value](#).

<i>Optional:</i>	yes
<i>Characteristics:</i>	for DropDownList and FixedValue , multilanguage is possible
<i>Possible values:</i>	any text
<i>Default value:</i>	
<i>Example:</i>	"Value": "Berlin", "DisplayValue": "Berlin - Germany"

"Value":

Actual value, in conjunction with [DisplayValue](#).

<i>Optional:</i>	yes
<i>Characteristics:</i>	for DropDownList and FixedValue
<i>Possible values:</i>	any text
<i>Default value:</i>	
<i>Example:</i>	"Value": "Berlin", "DisplayValue": "Berlin - Germany"

"Constraints": [see constraints](#)

4.2.2.2.2 Constraints

Use constraints to define:

- Conditions that must be fulfilled when entering the data
- [Creation rules](#)

The specification of constraints is optional.

If you define constraints for LDAP attributes, 8MAN checks whether the Active Directory also uses constraints for the attribute. If so, the stricter condition is applied. 8MAN shows in the [health check](#) which conditions are used.

Available constraints (all optional)

- **"MaxLength"**: maximum length. Default: -1 (unlimited).
- **"ForbiddenChars"**: Specifies which characters can not be used. Default: [] (empty list).
- **"ValidationRule"**: Regular Expression. Conditions that the entered text must meet.
- **"ValidationInformation"**: Tooltiptext, der bei Verletzung der Constraints angezeigt wird.
- **"UniquenessConstraint"**: **"properties/ldap/uniqueness"** Ensures the uniqueness of the input for AD attributes (prevents duplicates).
- **"CreationRule"**: A creation rule that automatically calculates and uses the value for the field. Only allowed if [DefaultValue](#) is not set.

Additional [validity checks and visibility controls](#) are available for Open Order Templates.

Example

```
"Constraints": {
  "MaxLength": 20
  "ForbiddenChars": ["ö", "ä", "ü", "ß"],
  "ValidationRule": "(?=[A-Z])",
  "ValidationInformation":
    "Use a maximum of 20 characters, no umlauts and at least one uppercase letter."
  "CreationRule": "<toLowerCase><firstLetter>({givenname}).{sn}",
}
```

4.2.2.2.3 MultiLanguageTemplates

Templates can be designed multilingual.

The language selected for the 8MAN login is used for the display. If there is no entry for the selected language, the first language is used.

Example

```
{
  "Key": "Name",
  "Value": {
    "Type": "TextField",
    "DefaultValue": "",
    "IsRequired": "true",
    "Label": "[ 'en-us:name', 'de-de:Name', 'fr-fr:Nom' ]"
  }
},
```

You can find more examples in the sample templates (.example) provided in the setup:

`%programdata%\protected-networks.com\8MAN\data\templates`

4.2.2.2.4 Creation rules

All input fields that can contain a constraints field can define a CreationRule within the constraints field, which automatically calculates the value of the field.

Creation rules are only valid if you do not define a default value.

Creation rules can be linked to one another as desired, e.g. „<firstLetter>({givenname}).{sn}@[fqdn]“. Spaces are also relevant.

The creation rule is also executed when the field:

- Is hidden ("IsHiddenFromRequester": true or "IsHidden": true)
- Is not editable ("IsEnabled": false)

Possibilities for creation rules

{sn}

This text is replaced by the current value of the input field for the LDAP attribute specified in curly braces (in this example, "sn").

This also works if the referenced input field is hidden and / or not editable.

If the referenced field contains a creation rule, it is executed first. The order of execution is calculated on the basis of such field dependencies. If the creation rules of a template form a cyclic field dependency (for example, if the creation rule for "sn" contains {cn} and that for "cn" {sn}), the template is rejected as invalid. The error is displayed in the [health check](#).

[fqdn]

This text is replaced by the domain name defined in the template ([FullQualifiedDomainName](#)).

Hello 123

Strings are accepted one by one, in this case "Hello 123".

The following special characters must be escaped with a backslash (\): backslash, round brackets, braces, comma.

Note: In JSON format, the double quotes and the backslash must be escaped with a backslash. Backslashes in creation rules must therefore be doubled, e.g.

- "\\(" for the round bracket
- "\\\" for a single backslash

A simple solution is provided by online tools that perform escaping for the JSON format e.g.

<http://www.infobyip.com/jsonencoderdecoder.php>. So you only have to manually perform the escaping for the creation rules.

<firstLetter>(…)

Returns the first character of the argument.

Example

<firstLetter>(Hello) is replaced by "H".

<toUpperCase>(…)

Converts the argument to uppercase.

Example

<toUpperCase>(Hello) is replaced by "HELLO".

<toLowerCase>(…)

Converts the argument to lowercase.

Example

<toLowerCase>(Hello) is replaced by "hello".

<trim>(…)

Deletes spaces at the beginning and end of the argument.

Example

<trim>(Hello) is replaced by "Hello".

<subst>(…)

Deletes blanks and hyphens from the argument, replacing letters with accents by letters or combinations of letters without accents.

Example

<subst>(Zoë Roßmäßler-Öker) is replaced by "ZoeRossmuesslerOeker".

<replace>(.,,..)**<replaceOnce>(.,,..)>**

Replaces characters.

Examples

<replace>(the dog and the fox,the,a) = "a dog and a fox"

<replaceOnce>(the dog and the fox,the,a) = "a dog and the fox"

<replace>(Norbert Van Eggert, ,) = "NorbertVanEggert"

<replace>(Norbert Van Eggert, ,) = "Norbert.Van.Eggert"

<regExpr>('...',...)

Specifies the first match of the regular expression (within the single quotation marks), applied to the second argument (begins immediately after the comma, spaces after the comma are counted).

Example

<regExpr>('{3}',Hello) Is replaced by "Hel".

All common regular expressions are supported. As a special feature, the grouping construct (? <This> ...) is also supported. The match on this group is returned.

Example

<regExpr>('{3}(?<this>.*)',Hello) Is replaced by "lo".

There are online tools that can be used to test regular expressions, e.g. <http://regex101.com>.

All functions can be arbitrarily nested.

Example

<regExpr>('{1}',<trim>(<toLowerCase>({sn})))

Complex example for an email address validation

```
"Name": "emailaddresses",
"Definition": {
  "Type": "TextArea",
  "Label": "Email addresses",
  "IsRequired": true,
  "IsEnabled": true,
  "Constraints": {
    "MaxLength": 500,
    "ValidationRule": "^((([a-z][a-z0-9]+)?([A-Z][A-Z0-9]+)?(\\w+([-+.']\\w+)*@\\w+([-.]\\w+)*\\.\\w+([-.]\\w+)*((\\r\\n)?\\n?)+)*$)",
    "ValidationInformation": "Does not match the email format!",
    "CreationRule" : "SMTP:<toLowerCase>({samaccountname})@<toLowerCase>([fqdn]
\\r\\nsmtp:<toLowerCase>(<firstLetter>({givenname})).<toLowerCase>({sn})@<toLowerCase>([fqdn])"
  }
}
```

4.2.2.2.4.1 LookupTable

With `LookupTable`, you create pairs of values that you use to fill fields automatically.

A definition for a lookup table has the following format:

- **LookupTableId**: This identifier is used to refer to the lookup table for additional fields.
- **LookupTable**: Value pairs of the table. The assignment is always one-to-one.

Example

In the following example, the user selects a company in a drop-down. Depending on the choice, the street, zip code, city are defined.

Define value pairs

```
"LookupTables": [
  { "Name": "LookupTableStreet",
    "Definition": {
      "Type": "LookupTable",
      "LookupTableId" : "Street",
      "LookupTable" : {
        "Demo Company Holding": "Demostreet 1",
        "Demo Company Marketing Solutions": "Demostreet 2",
        "Demo Company Services": "Demostreet 3"
      }
    }
  },
  { "Name": "LookupTableZIPcode",
    "Definition": {
      "Type": "LookupTable",
      "LookupTableId" : "ZIPcode",
      "LookupTable" : {
        "Demo Company Holding": "10000",
        "Demo Company Marketing Solutions": "20000",
        "Demo Company Services": "90000"
      }
    }
  },
  { "Name": "LookupTableCity",
    "Definition": {
      "Type": "LookupTable",
      "LookupTableId" : "City",
      "LookupTable" : {
        "Demo Company Holding": "Berlin",
        "Demo Company Marketing Solutions": "Hamburg",
        "Demo Company Services": "Munich"
      }
    }
  }
],
```

Drop down menu and fill the fields

```
{
  "Name": "company",
  "Definition": {
    "Type": "DropDownList",
    "Items": [
      { "Value": "Demo Company Holding", "DisplayValue": "Demo Company Holding" },
      { "Value": "Demo Company Marketing Solutions", "DisplayValue": "Demo Company Marketing
Solutions" },
      { "Value": "Demo Company Services", "DisplayValue": "Demo Company Services" }
    ],
    "Label": "Company"
  }
},
{
  "Name": "streetAddress",
  "Definition": {
    "Type": "TextField",
    "IsEnabled": false,
    "Constraints": {
      "CreationRule": "<lookup>(Street,{company})"
    },
    "Label": "Street"
  }
},
{
  "Name": "postalCode",
  "Definition": {
    "Type": "TextField",
    "IsEnabled": false,
    "Constraints": {
      "CreationRule": "<lookup>(ZIPcode,{company})"
    },
    "Label": "ZIP"
  }
},
{
  "Name": "l",
  "Definition": {
    "Type": "TextField",
    "IsEnabled": false,
    "Constraints": {
      "CreationRule": "<lookup>(City,{company})"
    },
    "Label": "City"
  }
},
}
```

4.2.2.2.5 Hide input fields

IsHiddenFromRequester

Specifies that the affected area is not displayed to the requester.

Optional: yes

Characteristics: Effective only in the web client / GrantMA, can be overridden by **"IsHidden":true**

Possible values: **true** or **false**

Default value: **false**

Example: **"IsHiddenFromRequester":true**

IsHidden

Specifies that the area is never displayed, even to the administrator in the post-processing of requests.

Optional: yes

Characteristics: if set to **true**, **IsHiddenFromRequester** is ineffective

Possible values: **true** or **false**

Default value: **false**

Example: **"IsHidden":true**

4.2.2.3 TextField

TextField is a single-line text input field.

Required

Type

Optional properties

Label

DefaultValue

Description

IsRequired

IsEnabled

Constraints

Example

```
{  
  "Type": "TextField",  
  "Label": "Text entry box 1",  
  "DefaultValue": "Apple",  
  "Description": "Please enter something.",  
  "IsRequired": true,  
}
```

4.2.2.4 TextArea

TextArea is a multi-line input field (for multi-line attributes). In [DefaultValue](#), line breaks (\r\n) may be included.

Required

[Type](#)

Optional properties

[Label](#)

[DefaultValue](#)

[Description](#)

[IsRequired](#)

[IsEnabled](#)

[Constraints](#)

Example

```
{  
  "Type": "TextArea",  
  "Label": "Multiline text input field 1",  
  "DefaultValue": "line1\r\nline2\r\nline3",  
}
```

4.2.2.5 MultiValueText

MultiValueText is a text input field for multiple values (for multi-value attributes).

Special conditions for **MultiValueText**:

- **DefaultValue** is a list of text values
- the **Constraints** are applied to each line

Required

Type

Optional properties

Label

DefaultValue

Description

IsRequired

IsEnabled

Constraints

Example

```
{  
  "Type": "MultiValueText",  
  "DefaultValue": [ "Apple", "Banana", "Orange" ],  
  "Label": "entry list",  
  "Description": "Please enter one or more values (one per line).",  
  "IsRequired": true,  
  "IsEnabled": true,  
  "Constraints": see Constraints  
}
```

4.2.2.6 DropDownList

A `DropDownList` is a selection list with non-editable values..

Required

[Type](#)
[Items](#)

Optional properties

[Label](#)
[DefaultValue](#)
[Description](#)
[IsRequired](#)
[IsEnabled](#)
[Constraints](#)

In addition, you define:

- **"Items"**: The list of values to select. A distinction is made here between **DisplayValue** (the value displayed in the selection list) and **Value** (the actual value that is stored for the LDAP attribute).
- **"DefaultValue"**: (optional) Specifies the value that is preselected when the template is loaded. This value must match a value in the Items list. Default value: The value of the first entry in Items.

Example

```
{
  "Type": "DropDownList",
  "DefaultValue": "Berlin",
  "Label": "Location",
  "Description": "Select the location of the user."
  "Items": [
    { "Value": "Berlin", "DisplayValue": "Berlin - Germany" },
    { "Value": "Vienna", "DisplayValue": "Vienna - Austria" }
  ]
}
```

4.2.2.7 FixedValue

FixedValue sets a fixed, non-editable value. The displayed value may differ from the value used.

Required

Type

Optional properties

Label

In addition, you define:

- **"DisplayValue"**: The value displayed in the template.
- **"Value"**: The actual value that is stored for the LDAP attribute.

Example

```
{  
  "Type": "FixedValue",  
  "Label": "Fixed value 1",  
  "Description": "This is a fixed Value."  
  "DisplayValue": "Displayed value",  
  "Value": "Real value"  
}
```

4.2.2.8 Checkbox

A checkbox, which knows the states activated/enabled (true) and deactivated/disabled (false).

Checkboxes are only used in the Modules area for the email activation of users (create a mailbox) and groups (create a distribution group) in Exchange.

Required

Type

Optional properties

Label

DefaultValue (valid values for checkboxes are only **true** or **false**)

Example

```
"Name": "createdistributiongroup",
"Definition": {
  "Type": "Checkbox",
  "DefaultValue": true,
  "Label": "Create distribution group",
}
```

4.2.2.9 Radio-Buttons

Radio is a group of radio buttons. You can only use radio buttons in Open Order Templates.

Properties

Type

For a group of radio buttons is the "Type": "Radio".

RadioGroupId

All radio buttons with the same id are grouped into one group. Within a group, only one radio button can be selected at a time.

Label

The value displayed in the form.

Value

The actual value that is stored.

IsChecked

Sets the initial selected radio button of a group.

Beispiel

```
"Key": "ActionRadio1",
"Value": {
  "Type": "Radio",
  "RadioGroupId": "Group1",
  "IsChecked": "true",
  "Label": "Displayed value 1",
  "Value": "Real value 1"
}
"Key": "ActionRadio2",
"Value": {
  "Type": "Radio",
  "RadioGroupId": "Group1",
  "Label": "Displayed value 2",
  "Value": "Real value 2"
}
```

4.2.2.10 AccountSearchTextField

AccountSearchTextField is an input option for searching for a user or group. A text field with an additional search button is displayed. If the button is pressed a search dialog appears. The search result can be further processed in the form.

AccountSearchTextField can only be used in Open Order Templates.

Properties

Type

"Type": "AccountSearchTextField"

Label

The annotation of the search field displayed in the form.

LookupTableId

Identifies the lookup table from which the LDAP attributes from the search can be used in the template.

AttributesToLoad

A list of LDAP attributes to load, for example ["sn", "cn"].

Example

// Define search input

```
{
  "Key": "Requester",
  "Value": {
    "Type": "AccountSearchTextField",
    "Label": "Request for",
    "LookupTableId": "RequesterSearchResult",
    "AttributesToLoad": [
      "sn",
      "givenname"
    ]
  }
},
// Use search results
{
  "Key": "given name",
  "Value": {
    "Type": "TextField",
    "Label": "given name",
    "IsEnabled": "false",
    "Constraints": {
      "CreationRule": "<lookup>(RequesterSearchResult,givenname)"
    }
  }
},
{
  "Key": "Surname",
  "Value": {
    "Type": "TextField",
    "Label": "Surname",
    "IsEnabled": "false",
    "Constraints": {
      "CreationRule": "<lookup>(RequesterSearchResult,sn)"
    }
  }
}
```

}

4.2.2.11 SearchField

SearchField ist ein Suchfeld.

Pflichtangaben

Type

ObjectType - Wonach gesucht wird, definieren Sie mit der Eigenschaft **ObjectType**. In der derzeitigen Entwicklungsstufe wird nur der Wert **Account** unterstützt.

optionale Properties

Label

DefaultValue

Description

IsRequired

IsEnabled

Constraints

Filters - gibt an, dass in der Technologie-Kategorie (zum Beispiel **ActiveDirectory**) nur Ergebnisse geliefert werden sollen, die der angegebenen Expression entsprechen (in dem Beispiel nur aktivierte User).

Beispiel

```
{
  "Key": "OwnerSearch",
  "Value": {
    "Type": "SearchField",
    "IsEnabled": "true",
    "Label": "['en-us:Owner', 'de-de:Besitzer']",
    "ObjectType": "Account",
    "Filters": {
      "ActiveDirectory": "PropertyValues('ObjectClass')->Value='user' AND (Status & 2) = 2"
    }
  }
},
```

4.3 Templates for users/groups/contacts

4.3.1 Customize templates for new users

4.3.1.1 Enter Name and OU

Create account within Active Directory [X]

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Vorname Nachname

Common-Name

Description Automatically generated description for .

SAM-Account-Name

Organisational unit (OU) Sales

LDAP Attributes

Group memberships

Start external program

Password options

User activation

Create an Exchange mailbox

Credentials 8man-demo\sa-8man

Please add a comment

Immediately

Cancel

Status of changes: ...

In the template shown, the first name ("givenname") and the last name ("sn") is mandatory.

The common name ("cn"), the SAM account name ("samaccountname") and the description are predefined by the creation rule.

Create account within Active Directory [X]

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Vorname Benjamin Nachname Button

Common-Name Benjamin Button

Description Automatically generated description for Benjamin Button.

SAM-Account-Name b.button

Organisational unit (OU) Sales

LDAP Attribute Marketing

Group memberships

Start external program

Password options

User activation

Create an Exchange mailbox

Credentials 8man-demo\sa-8man

Please add a comment

Immediately

Cancel

Status of changes: ...

To select the OU in which the new user is stored, this template uses a [DropDownList](#).

4.3.1.2 Enter additional LDAP attributes

In the LDAP Attributes area, further entries can be made for these. If you use an LDAP attribute in the assignment, 8MAN dynamically loads the attribute from the AD when loading the template. Please see: [Load a template](#).

In this example the "userprincipalname" is preset via [CreationRule](#) and [editable](#). The input field "company" is implemented as [DropDownList](#). Depending on the choice of the company, the "streetaddress", "postalcode" and the location ("1") are set. The user can [not edit](#) these values.

Example

```
"LdapAttributes": [
{
  "Name": "sn", //Assignment, the following definition for the input of the attribute "sn"
  "Definition": {
    /* property listing
    .
    . */
  }
}
]
```

4.3.1.3 Assign group memberships

Create account within Active Directory
✕

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Vorname Nachname

Common-Name

Description

SAM-Account-Name

Organisational unit (OU)

▼ LDAP Attributes

^ Group memberships

Accounts Templates Paste Clear

Name
<input type="checkbox"/> Marketing (8man-demo\Marketing)

i

The user will automatically become a member of the groups specified here.

You can either search for a group or select a group template.

▼ Start external program

▼ Password options

▼ User activation

Credentials [8man-demo\sa-8man](#)

Please add a comment

Status of changes: ...

In the group memberships area ("**Memberof**"), you can define in which groups the new user should become a member. Add the SIDs of the desired groups to the Accounts list.

Example

```
"Memberof": {
  "IsHiddenFromRequester": false,
  "IsHidden": false,
  "Accounts": [
    "sid:///ad/S-1-5-21-1545227963-2195427628-2857504096-1440"
  ]
},
```

4.3.1.4 Run an external program

Create account within Active Directory
✕

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Vorname	<input type="text" value="Benjamin"/>	Nachname	<input type="text" value="Button"/>
Common-Name	<input type="text" value="Benjamin Button"/>		
Description	<input type="text" value="Automatically generated description for Benjamin Button."/>		
SAM-Account-Name	<input type="text" value="b.button"/>		
Organisational unit (OU)	<input type="text" value="Sales"/>		
<div style="background-color: #eee; padding: 2px;"> ▼ LDAP Attributes </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Group memberships </div>			
<div style="background-color: #eee; padding: 2px;"> ▲ Start external program <div style="font-size: 0.8em; margin-top: 5px;"> <input type="checkbox"/> Start the program after the user creation The external program with the name Create a welcome package is located on 8MAN Server path \\srv-8man\scripts\WelcomePackage.ps1 Configured command line parameters are {CommonName} {samaccountname} {DomainName} </div> </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Password options </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ User activation </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Create an Exchange mailbox </div>			

Credentials [8man-demo\sa-8man](#)

Please add a comment

Status of changes: ...

In the Execute external program ("ScriptOptions") section, you can specify that a program (script) is executed after creating the new user.

Example

```
"ScriptOptions" : {
  "IsHiddenFromRequester": false,
  "IsScriptEnabledDefault": true,
  "DisplayName": "Create a welcome package",
  "Path": "\\srv-8man\scripts\WelcomePackage.ps1",
  "CommandLineParameters": "{CommonName} {samaccountname} {DomainName}"
},
```

4.3.1.5 Enter password options

Create account within Active Directory
✕

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Vorname <input type="text" value="Benjamin"/>	Nachname <input type="text" value="Button"/>
Common-Name <input type="text" value="Benjamin Button"/>	
Description <input type="text" value="Automatically generated description for Benjamin Button."/>	
SAM-Account-Name <input type="text" value="b.button"/>	
Organisational unit (OU) <input type="text" value="Sales"/>	
<input type="checkbox"/> LDAP Attributes	
<input type="checkbox"/> Group memberships	
<input type="checkbox"/> Start external program	
<input checked="" type="checkbox"/> Password options	
Initial password <input type="text" value="••••••••"/> <input checked="" type="checkbox"/> Hide password Generate a new password with a length of <input type="text" value="12"/> characters	
<input checked="" type="checkbox"/> The user must change the password at next logon <input type="checkbox"/> The user cannot change his password <input type="checkbox"/> The password never expires	
<input type="checkbox"/> User activation	
<input type="checkbox"/> Create an Exchange mailbox	

Credentials [8man-demo\sa-8man](#)

Please add a comment

Status of changes: ...

In the password options section, you specify how the initial password ("InitialPassword") and the password options ("PasswordOptions") are preset.

Example

```

"InitialPassword": {
  "MinLength": 12,
  "IsComplex": true,
  "IsMasked": false,
  "DefaultValue": "P@ssword1234",
  "Constraints": {
    "ValidationRule": "[^\\s]*",
    "ValidationInformation":
      "At least 12 characters, uppercase and lowercase letters, at least one digit or a special
      character. No spaces."
  }
},
>PasswordOptions": {
  "MustBeChangedAtNextLogonDefault": true,
  "CannotBeChangedByUserDefault": false,
  "NeverExpiresDefault": false
},

```

4.3.1.6 Activation options

Create account within Active Directory
✕

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Vorname	<input type="text" value="Benjamin"/>	Nachname	<input type="text" value="Button"/>
Common-Name	<input type="text" value="Benjamin Button"/>		
Description	<input type="text" value="Automatically generated description for Benjamin Button."/>		
SAM-Account-Name	<input type="text" value="b.button"/>		
Organisational unit (OU)	<input type="text" value="Sales"/>		
<div style="background-color: #eee; padding: 2px;"> ▼ LDAP Attributes </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Group memberships </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Start external program </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Password options </div>			
<div style="background-color: #eee; padding: 2px;"> ▲ User activation <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <div> <input checked="" type="radio"/> Activate immediately <input type="radio"/> Activate on 8/6/2017 12:00 AM <input type="radio"/> Do not activate </div> <div> <input type="checkbox"/> Account expires on 10/4/2017 12:00 AM </div> </div> </div>			
<div style="background-color: #eee; padding: 2px;"> ▼ Create an Exchange mailbox </div>			

Credentials [8man-demo\sa-8man](#)

Please add a comment

Immediately

Cancel

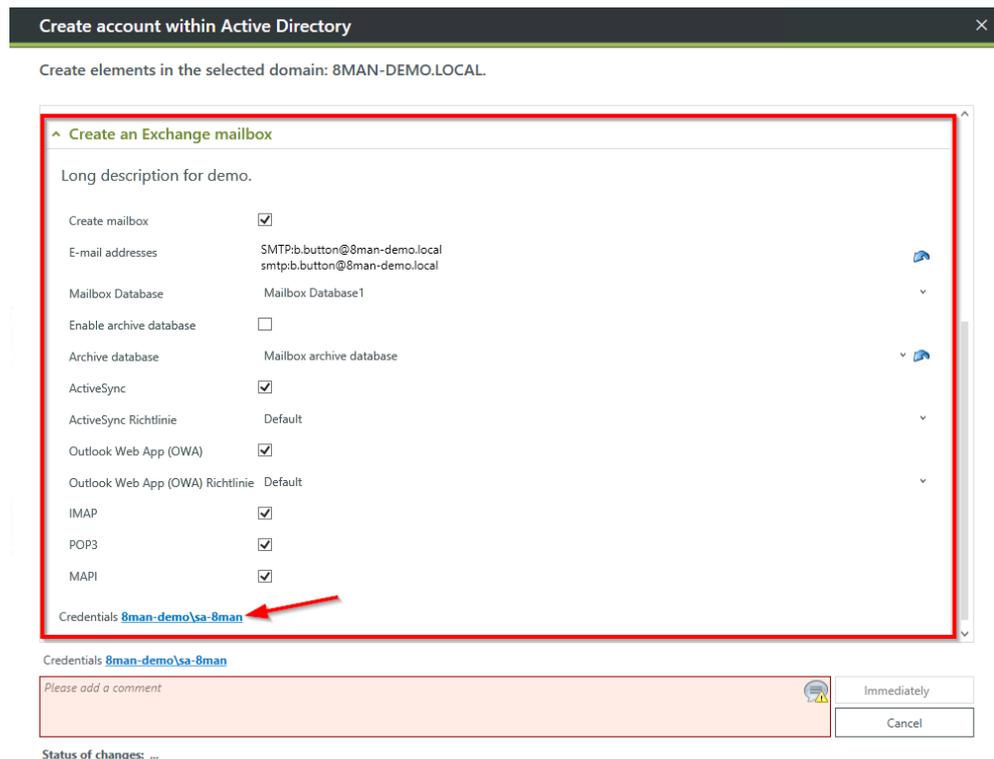
Status of changes: ...

In the Activation section, you determine whether the activation options ("ActivationOptions") are hidden.

Example

```
"ActivationOptions": {
  "IsHidden": false,
  "IsHiddenFromRequester": true
},
```

4.3.1.7 Create an Exchange mailbox



In the section "Modules", you enable the user to create an Exchange mailbox and to make settings for it in the same step. The entire area is optional and requires an 8MATE for Exchange license.

The credentials (arrow) can not be influenced by the template. You make this setting in the Exchange change configuration.

Example

```
"Modules" : [
{
  "Name" : "Exchange.Create.MailBox",
  "Displaytext" : "Create an Exchange mailbox.",
  "Description" : "Description text",
  "CredentialType" : "Windows",
  "Fields" : [
    {
      "Name": "createmailbox",
      "Definition": {
        "Type": "Checkbox",
        "DefaultValue": true,
        "Label": "Create mailbox"
      }
    },
    {
      "Name": "emailaddresses",
      "Definition": {
        "Type": "TextArea",
        "Label": "Email addresses",
        "IsRequired": true,
        "IsEnabled": true,
        "Constraints": {
          "MaxLength": 500,
          "ValidationRule": "^(((([a-z][a-z0-9]+)?([A-Z][A-Z0-9]+)?(\\w+([-+.']\\w+)*@\\w+([-.]\\w+)*\\.\\w+([-.]\\w+)*(@\\w+)?(\\r\\n)?\\n?)+)*)$",
          "ValidationInformation": "Does not match the Email format!",
          "CreationRule" : "SMTP:<toLowerCase>({samaccountname})@<toLowerCase>({fqdn})\\r\\nsmtp:<toLowerCase>(<firstLetter>({givenname})).<toLowerCase>({sn})@<toLowerCase>({fqdn})"
        }
      }
    }
  ],
  "Name": "MailboxDatabase",

```

```

"Definition": {
  "Type": "DropDownList",
  "DefaultValue": "Mailbox Database1",
  "Label": "Mailbox Database",
  "IsRequired": true,
  "Items": [
    {
      "Value": "Mailbox Database1",
      "DisplayValue": "Mailbox Database1"
    },
    {
      "Value": "Mailbox Database2",
      "DisplayValue": "Mailbox Database2"
    }
  ]
}
},
{
  "Name": "ActivateArchive",
  "Definition": {
    "Type": "Checkbox",
    "DefaultValue": "false",
    "Label": "Archive database"
  }
},
{
  "Name": "ArchiveDatabase",
  "Definition": {
    "Type": "DropDownList",
    "DefaultValue": "Mailbox Database1",
    "Label": "Archiv Datenbank",
    "IsRequired": true,
    "Items": [
      {
        "Value": "Mailbox Database1",
        "DisplayValue": "Mailbox Database1"
      },
      {
        "Value": "Mailbox Database2",
        "DisplayValue": "Mailbox Database2"
      }
    ]
  }
},
{
  "Name": "ActivateActiveSync",
  "Definition": {
    "Type": "Checkbox",
    "DefaultValue": "true",
    "Label": "ActiveSync"
  }
},
{
  "Name": "ActivateActiveSyncPolicy",
  "Definition": {
    "Type": "DropDownList",
    "DefaultValue": "Default",
    "Label": "ActiveSync policy",
    "IsRequired": true,
    "Items": [
      {
        "Value": "Default",
        "DisplayValue": "Default"
      },
      {
        "Value": "Other",

```

```

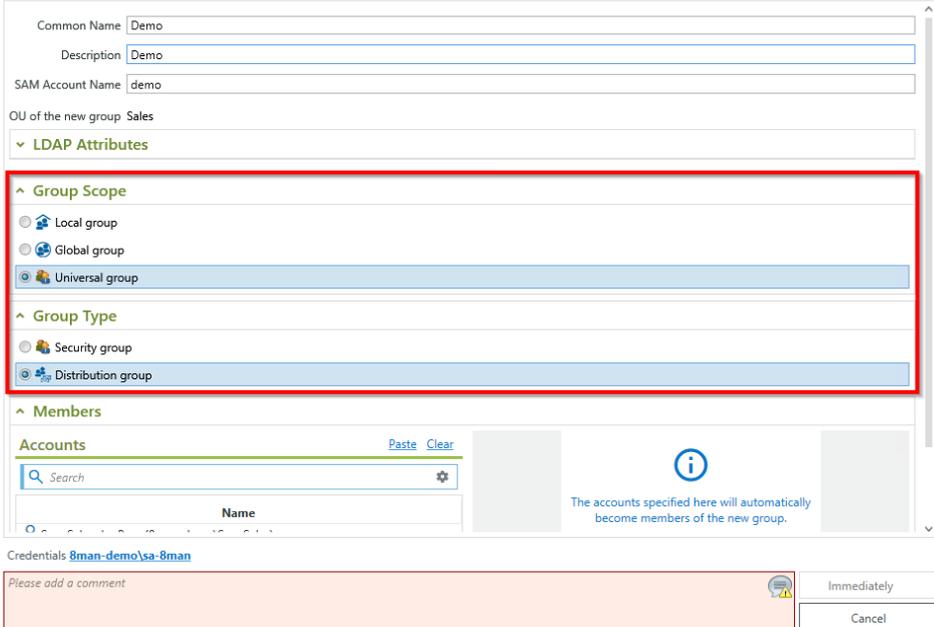
        "DisplayValue": "Other"
      }
    ]
  },
  {
    "Name": "ActivateOwa",
    "Definition": {
      "Type": "Checkbox",
      "DefaultValue": "true",
      "Label": "Outlook Web App (OWA)"
    }
  },
  {
    "Name": "ActivateOwaPolicy",
    "Definition": {
      "Type": "DropDownList",
      "DefaultValue": "Default",
      "Label": "Outlook Web App (OWA) policy",
      "IsRequired": true,
      "Items": [
        {
          "Value": "Default",
          "DisplayValue": "Default"
        },
        {
          "Value": "Other",
          "DisplayValue": "other"
        }
      ]
    }
  },
  {
    "Name": "ActivateImap",
    "Definition": {
      "Type": "Checkbox",
      "DefaultValue": "true",
      "Label": "IMAP"
    }
  },
  {
    "Name": "ActivatePop",
    "Definition": {
      "Type": "Checkbox",
      "DefaultValue": "true",
      "Label": "POP3"
    }
  },
  {
    "Name": "ActivateMapi",
    "Definition": {
      "Type": "Checkbox",
      "DefaultValue": "true",
      "Label": "MAPI"
    }
  }
]
}
]

```

4.3.2 Customize templates for new groups

The template for a new group contains many items that are also contained in the template for a new user. In the following, you will find only the different adjustments in a template for a new group.

4.3.2.1 Preset group options (group type/scope)



The screenshot shows the 'Create account within Active Directory' dialog. The 'Group Scope' section has three radio buttons: 'Local group', 'Global group', and 'Universal group'. The 'Universal group' option is selected. The 'Group Type' section has two radio buttons: 'Security group' and 'Distribution group'. The 'Distribution group' option is selected. A red box highlights the 'Group Scope' and 'Group Type' sections. Below these sections is a 'Members' section with a search box and a list of accounts. At the bottom, there are 'Immediately' and 'Cancel' buttons.

Specify which options are already preselected.

Example

"GroupTypeOptions": {

/ Determine the group scope radio button preset.*

Possible values:

- Global (Default)

- Local

*- Universal (must be used for email enabling) */*

"GroupArea" : "Universal",

/ Determine the group type radio button preset.*

Possible values:

- Security (Default)

*- Distribution (must be used for email enabling) */*

"GroupType" : "Distribution",

// Hide the area GroupTypeOptions to prevent user changes.

"IsHidden": false,

"IsHiddenFromRequester": false

},

4.3.2.2 Preset group members

Create account within Active Directory
×

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Common Name

Description

SAM Account Name

OU of the new group

▼ LDAP Attributes

▼ Group Scope

▼ Group Type

▲ Members

Accounts Paste Clear

⚙

Name
Sam Sales der Boss (8man-demo\Sam.Sales)

i

The accounts specified here will automatically become members of the new group.

You can search for users and groups.

▼ Create Distribution Group in Exchange

Credentials [8man-demo\sa-8man](#)

Please add a comment

Immediately
Cancel

Status of changes: ...

In the "members" section, define which members are already preset.

Example

```
"Members" : {
  "Accounts" :[
    "sid:///ad/S-1-5-21-2680840348-2237289205-2993809228-13534"
  ]
}
```

4.3.2.3 Enable e-mail (create distribution group) in Exchange

Create account within Active Directory
×

Create elements in the selected domain: 8MAN-DEMO.LOCAL

Common Name

Description

SAM Account Name

OU of the new group Sales

▼ LDAP Attributes

▼ Group Scope

▼ Group Type

▼ Members

^ Create Distribution Group in Exchange

Long description for demo.

Create distribution group

E-mail addresses SMTP:demo@8man-demo.local
smtp:demo@mydomain.com

Only authenticated senders

Credentials [8man-demo\sa-8man](#)

Credentials [8man-demo\sa-8man](#)

Please add a comment

Immediately
Cancel

Status of changes: ...

In the optional "Modules" section, you provide Exchange options. With this template, the new group can be email activated and become an Exchange distribution group. An 8MATE for Exchange license is required.

For a successful distribution role in Exchange, the group scope must be universal.

The credentials (arrow) can not be influenced by the template. You make this setting in the Exchange change configuration.

Example

```

"Modules" : [
{
// (required) Name of the module as key to depending processes. Do not change.
"Name" : "Exchange.Create.DistributionGroup",

// (required) Short description of the module. Used as section headline.
"Displaytext" : "Create Distribution Group in Exchange",

// (optional) Long description of the module. Displayed within the section.
"Description" : "Long description for demo.",

/* (optional) Determine the Credential Type.
Possible values:
- UsernamePassword (default)
- Windows */
"CredentialType" : "Windows",

// (required) required input values (all fields)
"Fields" : [
{
"Name": "createdistributiongroup",
"Definition": {
"Type": "Checkbox",
"DefaultValue": true,
"Label": "Create distribution group ",
"IsRequired": true
}
},
{
"Name": "emailaddresses",
"Definition": {
"Type": "TextArea",

```

```

    "Label": "E-Mail addresses ",
    "IsRequired": true,
    "IsEnabled": true,
    "Constraints": {
      "MaxLength": 500,
      "ValidationRule": "^((([a-z][a-z0-9]+)?([A-Z][A-Z0-9]+)?(\\w+([-+.' ]\\w+)*@\\w+([-.]\\w+)*\\.\\w+([-.]\\w+)*)(\\r\\n)?\\n?)+)*$",
      "CreationRule" : "SMTP:<toLowerCase>({samaccountname})@<toLowerCase>([fqdn])
\r\nsmtp:<toLowerCase>({samaccountname})@mydomain.com"
    }
  },
  {
    "Name": "RequireSenderAuthenticationEnabled",
    "Definition": {
      "Type": "Checkbox",
      "Label": "Only authenticated senders ",
      "DefaultValue": true,
      "IsRequired": true
    }
  }
]
}
]

```

4.3.3 Customize templates for new contacts

With customized templates for contacts, you can create contacts with 8MAN.

Templates for new contacts record the following three values:

- OU (organizational unit)
- Name
- Email address

You can only use templates for contacts in the Rich Client (not the Web client).

In order to display all information from contacts in 8MAN, you must configure the AD scan ([see knowledgebase article](#), Login required) and the Exchange-Scan accordingly.

example

```
[
{
  "TemplateType": "CreateMailContact",
  "Version": 1,
  "Id": "2adee521-9423-464e-a52b-0d20a54ec4f6",
  "DisplayName": "Create contact",
  "Description": "Creates a contact with Exchange",
  "FullQualifiedDomainName": "8man-demo.local",

  "OrganizationalUnit": {
    "Definition": {
      "Type": "DropDownList",
      "Items": [
        {
          "Value": "OU=Sales,OU=Berlin,DC=8man-demo,DC=local",
          "DisplayValue": "Sales"
        },
        {
          "Value": "OU=Marketing,OU=Berlin,DC=8man-demo,DC=local",
          "DisplayValue": "Marketing"
        }
      ],
      "DefaultValue": "OU=Sales,OU=Berlin,DC=8man-demo,DC=local",
      "Label": "Organizational unit (OU)"
    }
  },

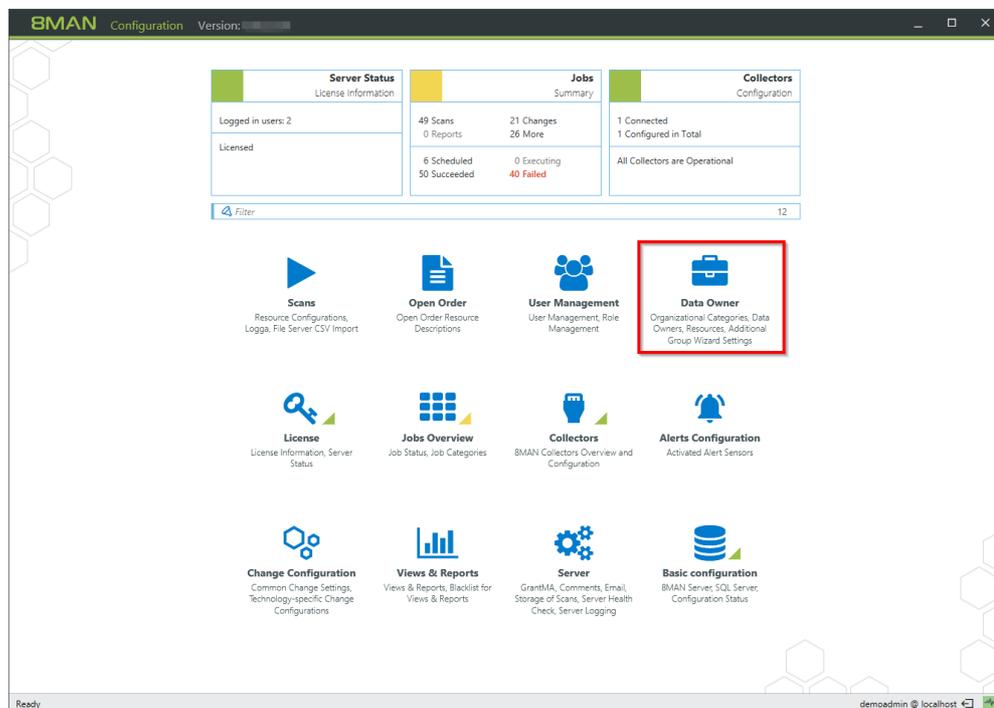
  "LdapAttributes": [
    {
      "Name": "name",
      "Definition": {
        "Type": "TextField",
        "Label": "Name",
        "IsRequired": true,
        "IsEnabled": true,
        "IsHidden": false,
        "Constraints": {
          "MaxLength": 50
        }
      }
    },
    {
      "Name": "externalemailaddress",
```

```
"Definition": {
  "Type": "TextField",
  "Label": "External email address",
  "IsRequired": true,
  "IsEnabled": true,
  "Constraints": {
    "MaxLength": 200,
    "ValidationRule": "[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}"
  }
}
]
```

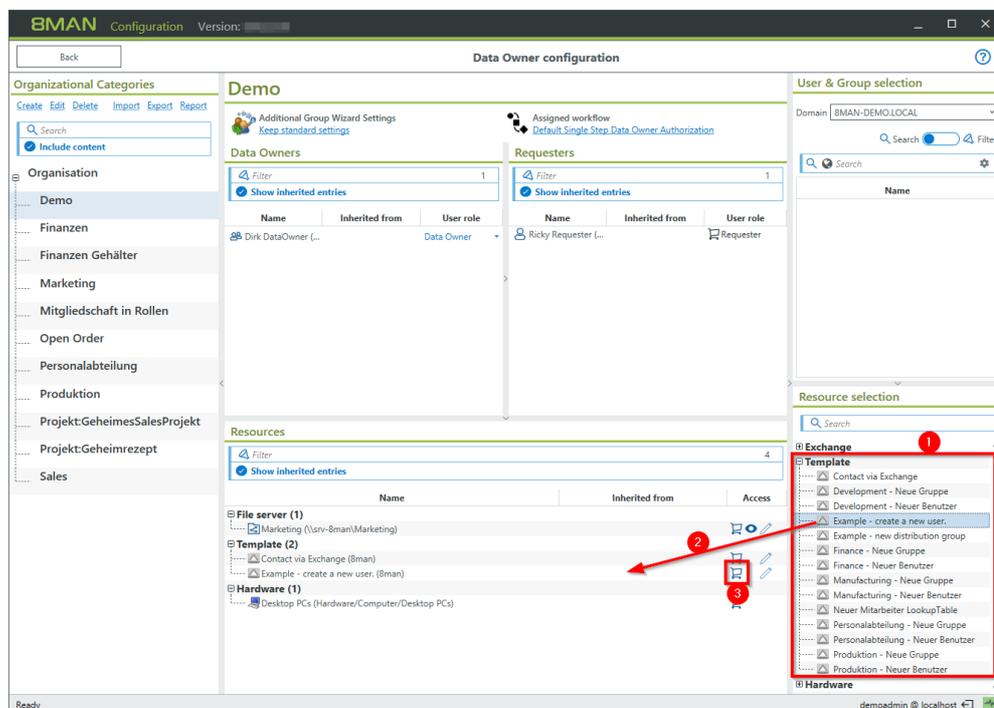
4.3.4 Make templates for users/groups/contacts available in the Web client

Templates for users/groups/contacts are available for use in the rich client as soon as they have been successfully loaded (see [Load templates](#)).

In order to allow a "requester" in the 8MATE GrantMA to use a template, the template must be assigned to an organizational category as a resource.



In the 8MAN configuration, start the Data Owner configuration.



1. *Successfully loaded* custom templates are automatically displayed in the resource selection.
2. Drag and drop a template into the resources area.
3. The template is automatically set to "Resource can be requested".

4.4 Open Order templates

Open Order templates differ from the templates for users / groups / contacts by the following characteristics:

- OpenOrder templates can only be used in the GrantMA / web client.
- Open Order Templates can be used for a wide range of orders. Therefore, there are no specialized input options or modules, but only freely definable containers, which can be interleaved as often as required.
- Open Order templates are assigned to Open Order resources via an XML configuration file. After uploading the XML configuration, only the resources are displayed in the data owner configuration, not the corresponding templates.

4.4.1 Structure of an Open Order template

The required information in the [header of the template](#) is the same as for users / groups / contacts.

The structure of the input form follows the following scheme:

```
"Form": {
  "Type": "Container",
  "Label": "Labeling",
  "Templates": [
    {
      "Key": "Value1",
      "Value": {
        "Type": input method
      }
    },
    { "Key": "^Value2",
      "Value": {
        "Type": input method
      }
    }
  ]
}
```

As an input method, you can use containers to create nesting.

With **CollapsibleContainer**, you create a container that can be collapsed and unfolded, using the **IsCollapsed** property to set the default.

Example for nested containers:

```
"Form": {
  "Type": "Container",
  "Label": "Root container",
  "Templates": [
    {
      "Key": "nested_container",
      "Value": {
        "Type": "CollapsibleContainer",
        "Label": "Nested container",
        "IsCollapsed": true,
        "Templates": [
          {
            "Key": "Collapsible_grandchild_container",
            "Value": {
              "Type": "Container",
              "Label": "Additional container",
              "Templates": [
                {
                  "Key": "Container3",
                  "Value": {
                    "Type": "CollapsibleContainer",
                    "Label": "And one more to fold",
                    "Templates": [
                      { //etc...
                    ]
                  }
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

4.4.2 Create an input option

The same input options are available for Open Order Templates as for templates for users, groups, and contacts. You can also use the same constraints and creation rules.

There are additional possibilities for inputs, descriptions, visibility control and validity checks, which can only be used in Open Order templates in the Web client (not in the Rich Client).

An overview of available input options can be found [here](#).

4.4.3 Specific Open Order input options

4.4.3.1 AccountSearchTextField

AccountSearchTextField is an input option for searching for a user or group. A text field with an additional search button is displayed. If the button is pressed a search dialog appears. The search result can be further processed in the form.

AccountSearchTextField can only be used in Open Order Templates.

Properties

- Type**
"Type": "AccountSearchTextField"
- Label**
The annotation of the search field displayed in the form.
- LookupTableId**
Identifies the lookup table from which the LDAP attributes from the search can be used in the template.
- AttributesToLoad**
A list of LDAP attributes to load, for example ["sn", "cn"].

Example

```
// Define search input
{
  "Key": "Requester",
  "Value": {
    "Type": "AccountSearchTextField",
    "Label": "Request for",
    "LookupTableId": "RequesterSearchResult",
    "AttributesToLoad": [
      "sn",
      "givenname"
    ]
  }
},
// Use search results
{
  "Key": "given name",
  "Value": {
    "Type": "TextField",
    "Label": "given name",
    "IsEnabled": "false",
    "Constraints": {
      "CreationRule": "<lookup>(RequesterSearchResult,givenname)"
    }
  }
},
{
  "Key": "Surname",
  "Value": {
    "Type": "TextField",
    "Label": "Surname",
    "IsEnabled": "false",
    "Constraints": {
```

```
    "CreationRule": "<lookup>(RequesterSearchResult,sn)"  
  }  
}
```

4.4.3.2 Radio-Buttons

Radio is a group of radio buttons. You can only use radio buttons in Open Order Templates.

Properties

Type

For a group of radio buttons is the "Type": "Radio".

RadioGroupId

All radio buttons with the same id are grouped into one group. Within a group, only one radio button can be selected at a time.

Label

The value displayed in the form.

Value

The actual value that is stored.

IsChecked

Sets the initial selected radio button of a group.

Beispiel

```
"Key": "ActionRadio1",
"Value": {
  "Type": "Radio",
  "RadioGroupId": "Group1",
  "IsChecked": "true",
  "Label": "Displayed value 1",
  "Value": "Real value 1"
}
"Key": "ActionRadio2",
"Value": {
  "Type": "Radio",
  "RadioGroupId": "Group1",
  "Label": "Displayed value 2",
  "Value": "Real value 2"
}
```

4.4.6 Open Order Vorlagen in den 8MATE GrantMA einbinden

To create Open Order Templates, follow these steps:

1. [Enter the template's call into the XML Resource Configuration](#)
2. [Upload an XML resource configuration to the Data Owner configuration](#)
3. [Set the Open Order resource to requestable](#)

4.4.6.1 Enter the template's call into the XML Resource Configuration

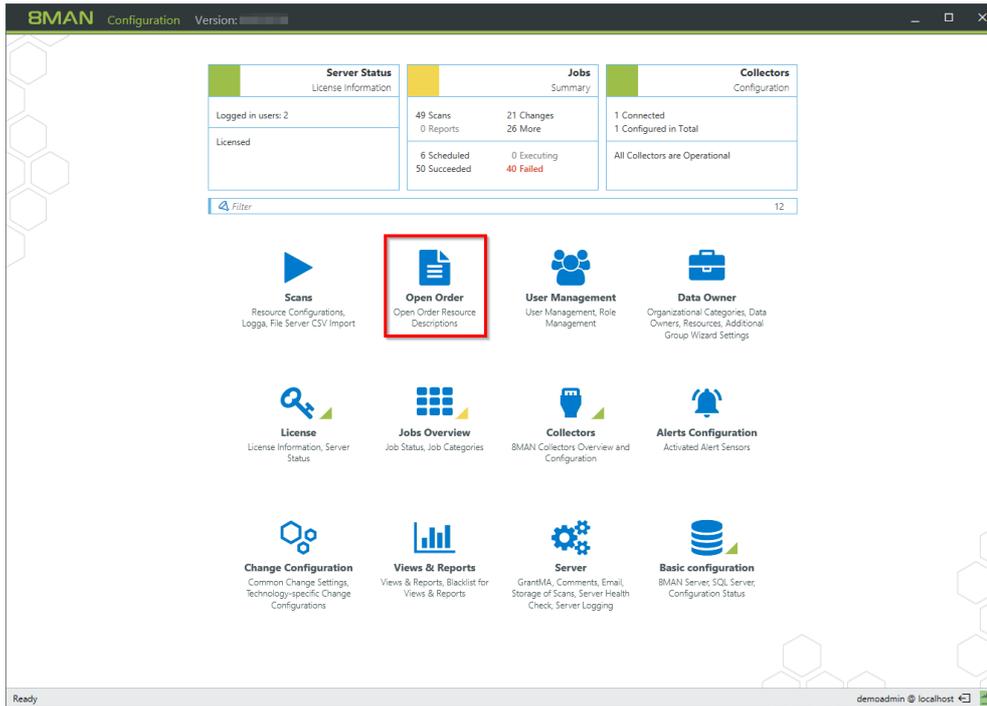
Assign the [unique ID](#) of the OpenOrderTemplate to one or more resources.

For more information on the structure of the XML resource configuration, see the Open Order manual.

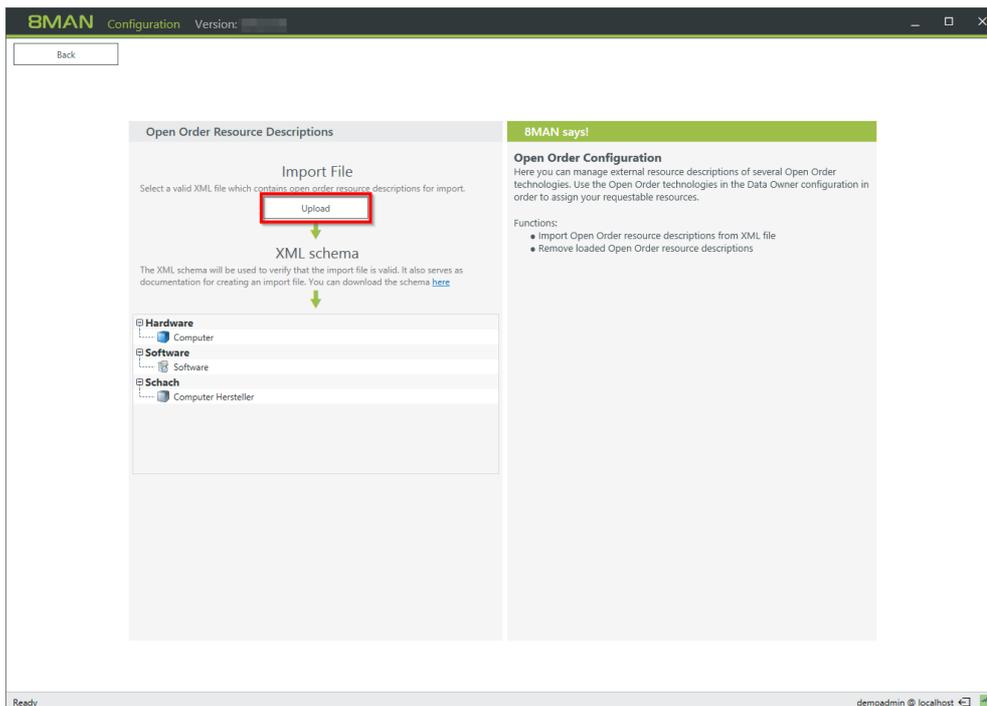
Example

```
<?xml version="1.0" encoding="utf-8"?>
<resourceImport Version="3">
  <technology Id="D54C16F2-42C1-477A-BD20-3285158F68D3" Name="Hardware" IconId="2" Color="#0000be">
    <definitions>
      <permissionSets>
        <permissionSet PermissionSetId="1" Description="['en-US:Buy','de-DE:Kaufen']" />
        <permissionSet PermissionSetId="2" Description="['en-US:Lease','de-DE:Leasen']" />
        <permissionSet PermissionSetId="3" Description="['en-US:Rent','de-DE:Mieten']" />
      </permissionSets>
      <types>
        <type Id="1" Description="['en-US:Hardware','de-DE:Hardware']" IconId="Container"
PermissionSetIds="[]" />
        <type Id="3" Description="['en-US:Desktop','de-DE:Desktop']" IconId="Computer"
PermissionSetIds="[1,2,3]" />
      </types>
    </definitions>
    <data>
      <root Id="6CE9B526-9FFD-46A5-9ED0-36FB4E1303B5" Name="Computer" TypeId="1" Merge="no">
        <resource Name="Desktop PCs" TypeId="3" Description="['en-US:Stationary PC','de-DE:Stationäre
Arbeitsplatz-PCs']">
          <resource Name="Desktop-PC Simple" TypeId="3" />
          <resource Name="Desktop-PC Standard" TypeId="3" />
          <resource Name="Desktop-PC Custom" TypeId="3" TemplateID="E3865726-6FDF-489E-A7D5-
4ABBA5B2BF83" />
        </resource>
      </root>
    </data>
  </technology>
</resourceImport>
```

4.4.6.2 Upload an XML resource configuration to the Data Owner configuration

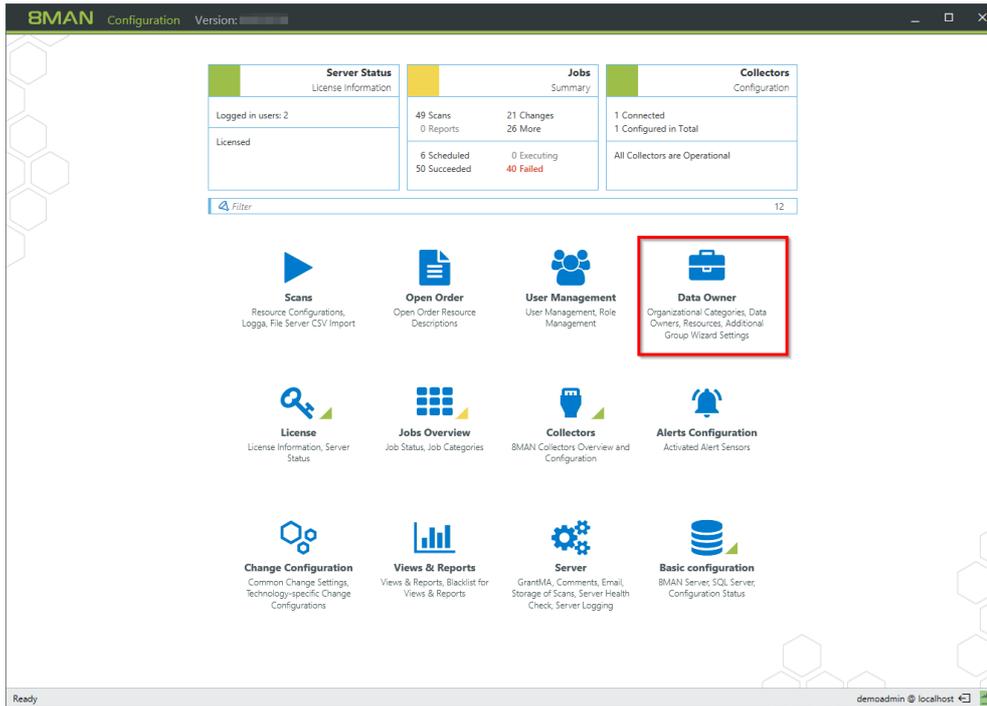


In the 8MAN configuration, click "Open Order".

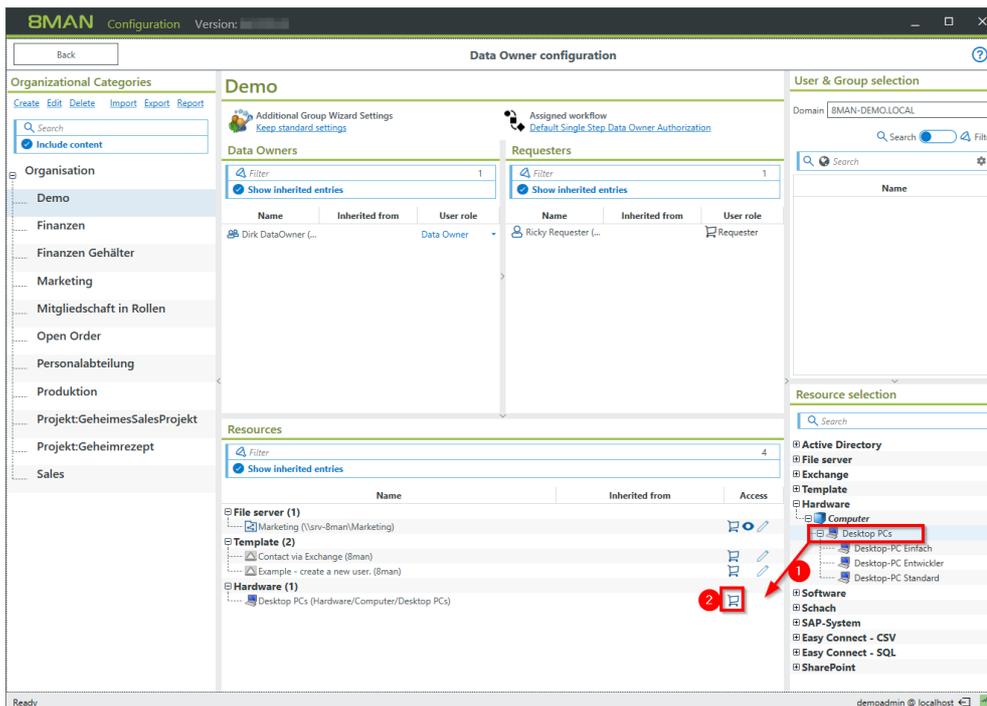


Click "Upload" to import the XML Resource Configuration. After successful import, the resources are available in the Data Owner configuration and can be assigned to organizational categories.

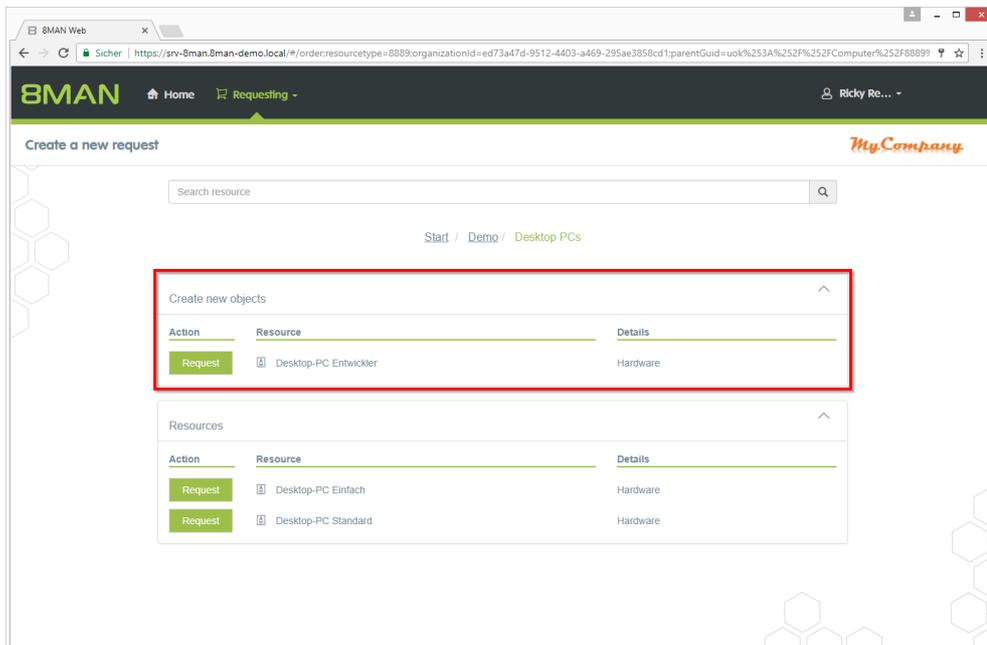
4.4.6.3 Set the Open Order resource to requestable



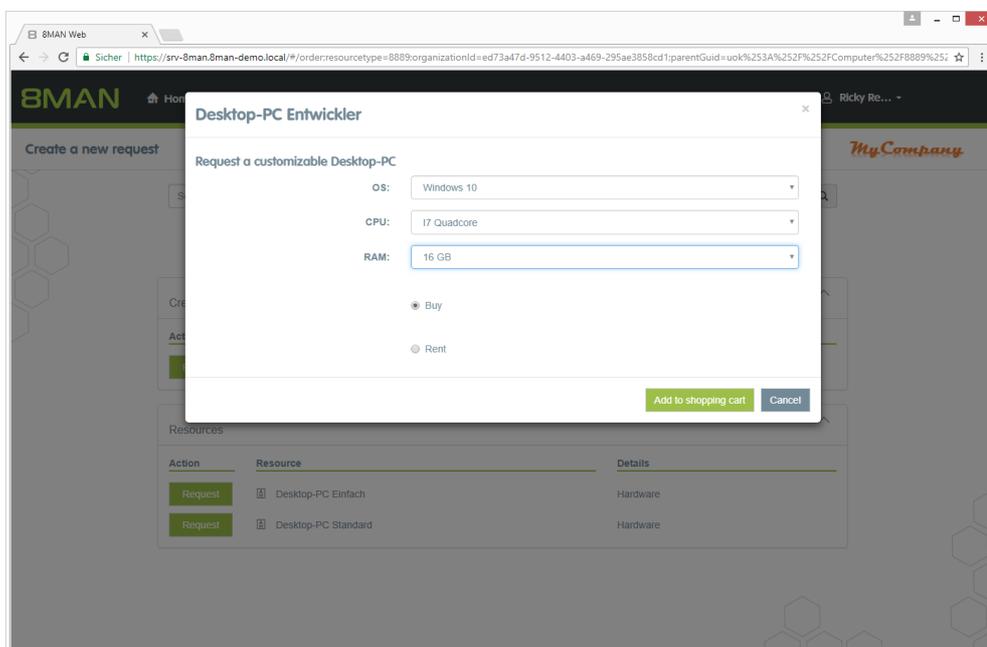
In the 8MAN configuration, click "Data Owner".



1. Add the desired resource by drag & drop.
2. The resource is automatically marked as requestable.



The requester can find the resource available via Open Order in the "Create new objects" area.



Example for an template based Open Order request.