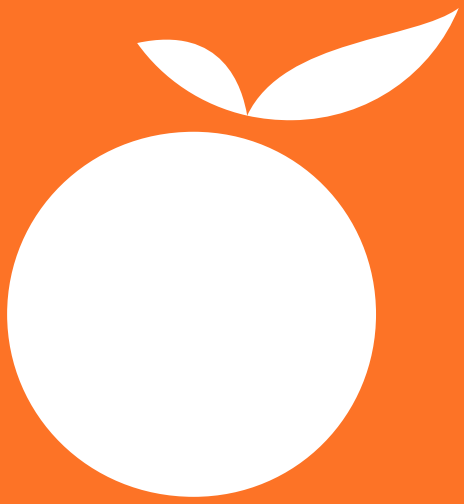
tts performance suite



Manual

Importing structures from Excel

Imprint

Importing structures from Excel
26. March 2021

Table of Contents

1 Excel import - the basics

Excel imports are carried out with the help of two files:

- a configuration file (containing the necessary technical information)
- and an Excel file (containing the necessary content-related information).

Each row of the Excel file contains a main dataset (process, document or user) that is about to be imported, as well as any additional information regarding secondary objects. This could involve maintenance roles, (process) roles or organizational units, for example.

If they don't already exist, secondary objects are generated and - where possible - automatically assigned to the main object.
Extractors are used for both the read-out and for interpreting the data. They are configured via the configuration file for the import.

1.1 Column extractors

There are three types of extractors:

- **Untyped extractors for object-specific attributes**

```
<extractor name="AnyName">
    <set-parameter name="column" value="A"/>
    <set-parameter name="property" value="language"/>
</extractor>
```

These extractors simply read-out the value at the corresponding position. It is, however, also possible to link them with value mappers (described below).

- **Typed extractors (**technicalName, entityValueList, propertyValue**)**

```
<extractor name=" AnyName " type="technicalName">
    <set-parameter name="column" value="B"/>
</extractor>
```

These extractors are used in various imports.

- **Specific extractors (**processStructure,processLevel**)**

These extractors are import-specific and perform specific tasks.

1.2 Other configuration elements

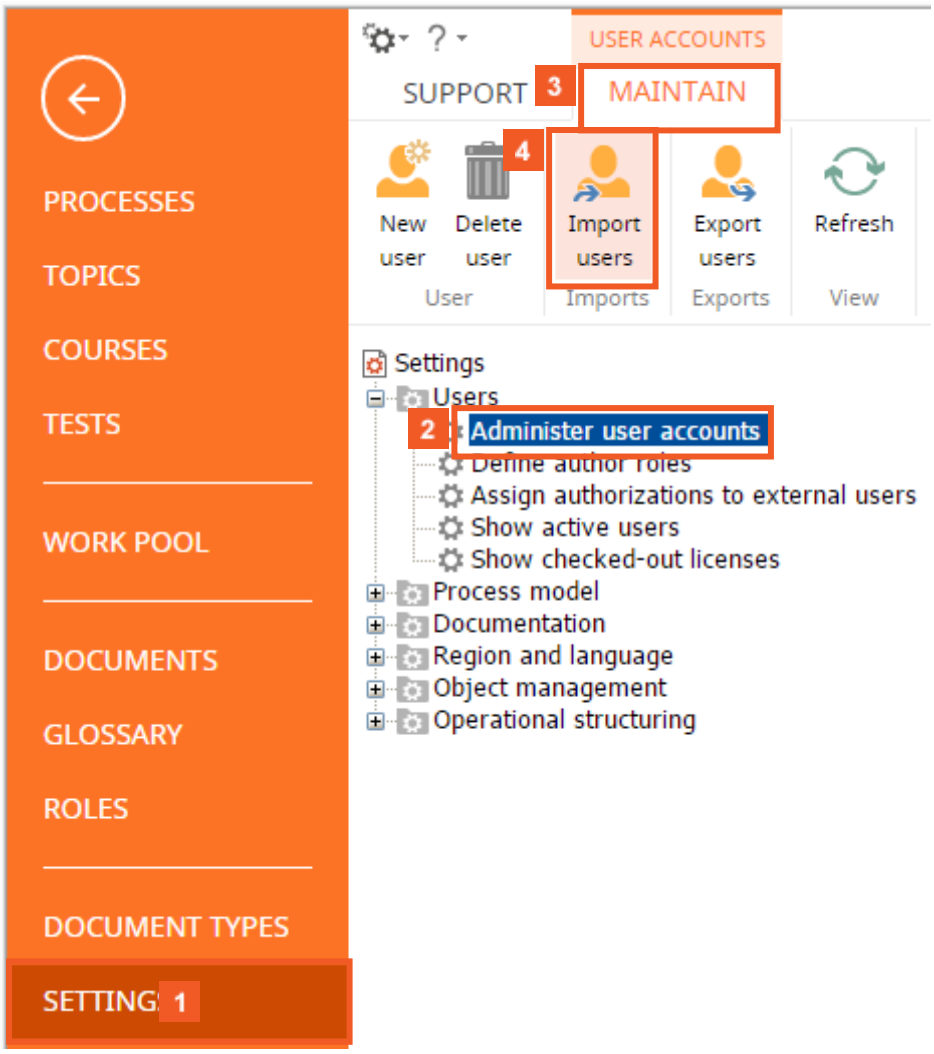
- The **value-mapping** construct**:**
  value mappers can be used within the extractors. They merely replace values in the Excel file with substitute values.
  The **value-mapping** construct**:**
  value mappers can be used within the extractors. They merely replace values in the Excel file with substitute values.

```
<value-mapping>
    <mapping importValue="Global" systemValue="area"/>
    <mapping importValue="Prozess" systemValue="process"/>
    <mapping importValue="Prozessschritt" sys-
temValue="processstep"/>
</value-mapping>
```

This function is particularly useful when one wants to replace abstract technical names
with *human-readable* representations in the Excel sheets.

2 How to import user accounts and their settings from Excel

You can easily import user accounts and their settings from Excel. To do this:

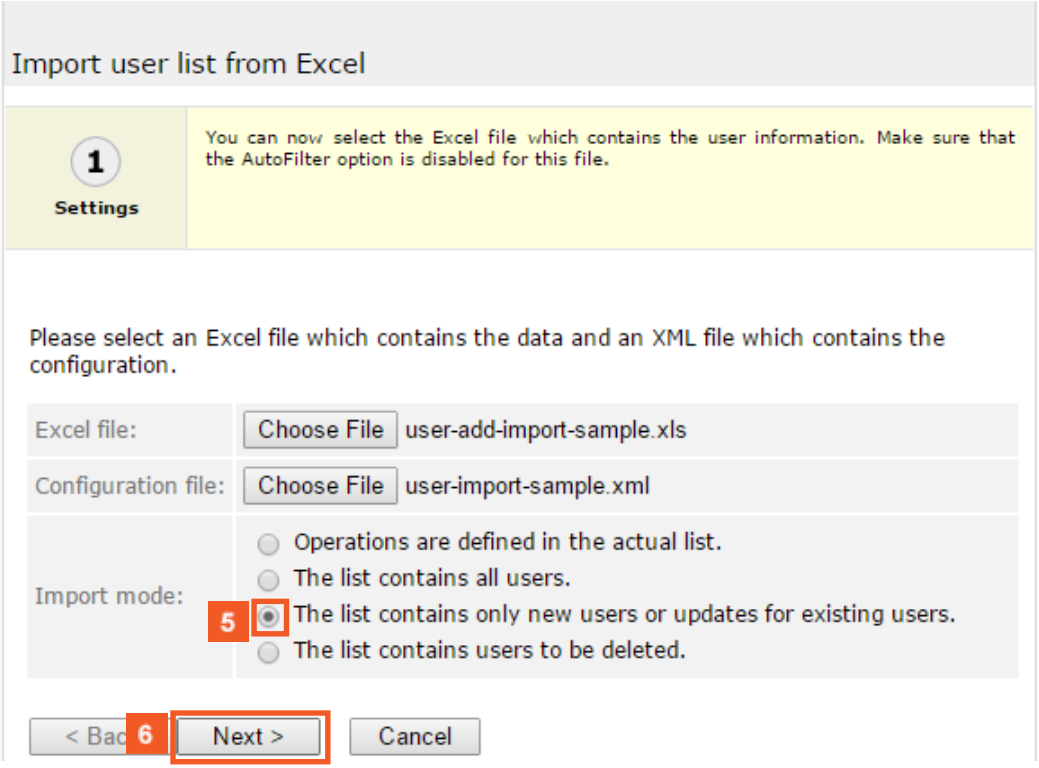


1 Click the **Settings** tab.

2 Click the **Administer user accounts** tree item.

3 Click the **Maintain** tab.

4 Click the **Import users** button.

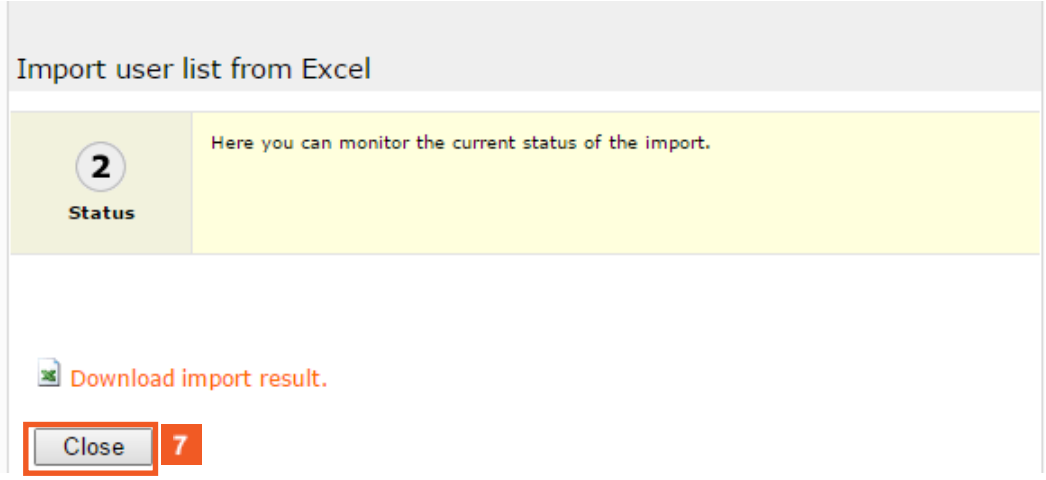


When performing a user Excel import, you can choose from the following options:

- Operations are defined in the actual list.
  When this is set, it is expected that the operation that is to be applied to the users (create, delete or update) is stored at each main dataset within the Excel file (please refer to the "Option" section under untyped extractors).
- The list contains all users.
  Existing users, who are not in the Excel sheet, will be deleted.
- The list contains only new users or updates for existing users.
  Only new users will be added, or existing users will be updated.
- The list contains users to be deleted.
  All users referred to in the list will be removed.

5 Click the **The list contains only new users or updates for existing users** radiobutton.

6 Click the **Next >** button.



7 Click the **Close** button.

3 User import via Excel: Configuration

General information on the user Excel import

When performing a user Excel import, you can choose from the following options:

- Operations are defined by the list itself.

When this is set, it is expected that the operation that is to be applied to the users (create, delete or update) is stored

at each main dataset within the Excel file. (please refer to the "Option" section under untyped extractors).

- The list contains all users.

Existing users, who are not in the Excel sheet, will be deleted.

- The list contains only new users or updates for existing users.

Only new users will be added, or existing users will be updated.

- The list contains users to be deleted.

All users referred to in the list will be removed.

Each row of the Excel file contains a main dataset (a user) as well as optional secondary datasets for roles and/or author roles.

Both roles and author roles can be created with this import.

Roles and author roles are automatically assigned to the user (main dataset).

General parameters

Name	Value	Description
sheet	Examples: "sample" ("sample" sheet) "1" (1st sheet)	Name of the Excel sheet on which the import information is found, or the sheet no. (beginning with 1)
dataBeginRow	Integer value beginning with 1	Indicates the row where the data begins. "2", for example, would mean that the data starts on the second row. The first row can then be used for column headers.

Examples from the configuration file:

```
<!-- first two rows may be used as the header -->
<set-parameter name="dataBeginRow" value="3"/>
<!-- name of the Excel sheet to be used -->
<set-parameter name="sheet" value="sample"/>
```

The following untyped user import-specific extractors can be used.

Name	Value range	Description
operation	add, delete, update	Indicates whether the user is to be added, deleted or updated.
licgroup	The license group to be primarily used	Name of the primary license group
fblicgroup	The license group to be used secondarily	Name of the secondary (*fallback*) license group
checkoutlic	0,1	0 = User does not have the right to use concurrent licenses. 1 = User has the right to use concurrent licenses.
authmode	intern, extern, both	intern = Login via the Workbench extern = Login via LDAP both = User can use either of the two login variants
password	User's password	Password in plain text
ldapServer	ldap server ip and port e.g. ldapserverdomain:389	Is used by the Simple LDAP Login module to facilitate authentication via an LDAP server.
ldapUserDN	LDAP distinguished name of the user	The unambiguous LDAP name that is to be used to identify the user.
firstname	The user's first name	
lastname	The user's family name	
email	The user's e-mail address	
editLanguage	e.g. de-de, en-us, fr-fr	Editing language in RFC1766 (the default editing language will be used if this is not configured)
uiLanguage	e.g. de, en, fr	User interface language in ISO639 (the default user interface language will be used if this is not configured)

dialogSize	small, normal, big	Default normal
clipLength	Integer that indicates the length of the where-used list	Default 15

Parameters	Function
technicalName (mandatory)	Extracts the user's technical name. This is identical to the user's login name and is therefore necessary.
entityValueList (optional)	Extracts technical names from a comma-separated list of the secondary objects (roles / author roles) that are to be assigned. The roles / author roles will be created if they do not yet exist, and they will be automatically assigned to the user.

The following typed extractors can be used during the user import.

Parameters	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated

Available parameters for the technicalName extractor:

Mode of operation:

As an input value, the **technicalName** extractor is given a column from which the user's username is to be read-out.

Example from the configuration file:

```
<extractor name="name" type="technicalName">
    <set-parameter name="column" value="B"/>
</extractor>
```

Example from the Excel file:

	A	B	C	D	E	F	G	H	I	J
1	Operation	Login name	Password	Auth-mode	Primary license group	Allow license checkout for concurrent licenses	First name	Family name	UI language	Editing language
2	add	tester1	tester1	Both	11111	1	Umberto	Eco	de	de-de
3	add	tester2	tester2	Intern	11112	1	Dan	Brown	de	de-de
4	add	tester3	tester3	extern	11113	0	Tom	Clancy	en	en-us

Parameters	Value range	Description
column	A-ZZ	Excel identifier for the column to be evaluated, with the titles of the roles or author roles. Titles are displayed in a comma-separated list here.
titleLanguage	Language code (RFC1766) e.g. de-de	To be used with the "column" parameter. This indicates the language to be used to search for a role / author role with the

		specified title.
idColumn	A-ZZ	Excel identifier for the column to be evaluated in which the technical names of the role / author role are found. Technical names are displayed in a comma-separated list here.
targetEntity	role, mrole	Indicates that objects of the (process) roles or author roles types are to be assigned. They will be created if they don't already exist.

Available parameters for the entityValueList extractor:

Mode of operation:

The **entityValueList** extractor supports two different ways of searching for existing objects.

- Via the technical names. The extractor is given a column - via the "idColumn" parameter - in which a comma-separated list of the technical names for roles / author roles is expected.
- Via the title of the object. The extractor is given a comma-separated list of titles for roles / author roles - via the "column" parameter - as well as the language, via the "titleLanguage" parameter. There is then a search for existing objects with these titles in the appropriate language.

If both exist, the search for existing objects always takes precedence over the technical names.

Roles and author roles that do not already exist will be created.

Roles and author roles are automatically assigned to the user (main dataset).

It's possible that several extractors of this type are found on the same row - e.g. an extractor for roles and an extractor for author roles.

It is currently not possible to hierarchically subdivide imported roles in the tree structure. All objects are created on the top level of the Role tree.

If technical names are indicated - but not the titles - the technical name will be used as the object title.

If the titles are indicated - but not the technical names - the technical name will be filled with the title (in a modified form if necessary).

Even when the "titleLanguage" has been indicated, the title will still always be created in the user's current editing language.

	A	B	C	D	E
1	Process title on level 1	Process title on level 2	Process title on level 3	Process level no.	Technical name of the process
2	Process 1 - Level 1			Global	11111
3	Process 1 - Level 1	Process 2 - Level 2		Process	11112
4	Process 1 - Level 1	Process 2 - Level 2	Process 3 - Level 3	Process step	11113
5	Process 1 - Level 1	Process 2 - Level 2	Process 4 - Level 3	Process step	11114

Example from the configuration file (assignment via the technical names of the process roles):

```
<!-- match roles using techn. names -->
 <extractor name="Roles" type="entityValueList">
        <set-parameter name="idColumn" value="L" />
        <set-parameter name="targetEntity" value="role" />
 </extractor>
<!-- match mroles using techn. names and author role titles in English -->
 <extractor name="MRoles" type="entityValueList">
        <set-parameter name="column" value="K"/>
        <set-parameter name="titleLanguage" value="en-us"/>
        <set-parameter name="idColumn" value="K" />
        <set-parameter name="targetEntity" value="mrole" />
 </extractor>
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	Operation	Login name	Password	Auth-mode	Primary li-cense group	Checkout licenses	First name	Family name	UI lan-guage	Editing language	Maintenance roles - technical name	Roles - tech-nical name
2	add	tester1	tester1	both	11111	1	Umberto	Eco	de	de-de	admin	
3	add	tester2	tester2	intern	11112	1	Toni	Morrison	de	de-de	mrole1,Author	
4	add	tester3	tester3	extern	11113	0	Tom	Clancy	en	en-us	mrole1	role1,role2

Important:

The order in which the extractors appear within the configuration file is important.

The following order should be used:

- Operation (specific attribute)
- technicalName (Login name)
- Other user-specific attributes
- entityValueList assignments to roles / author roles.

Examples:

Sample files can be obtained from the tts docsportal.

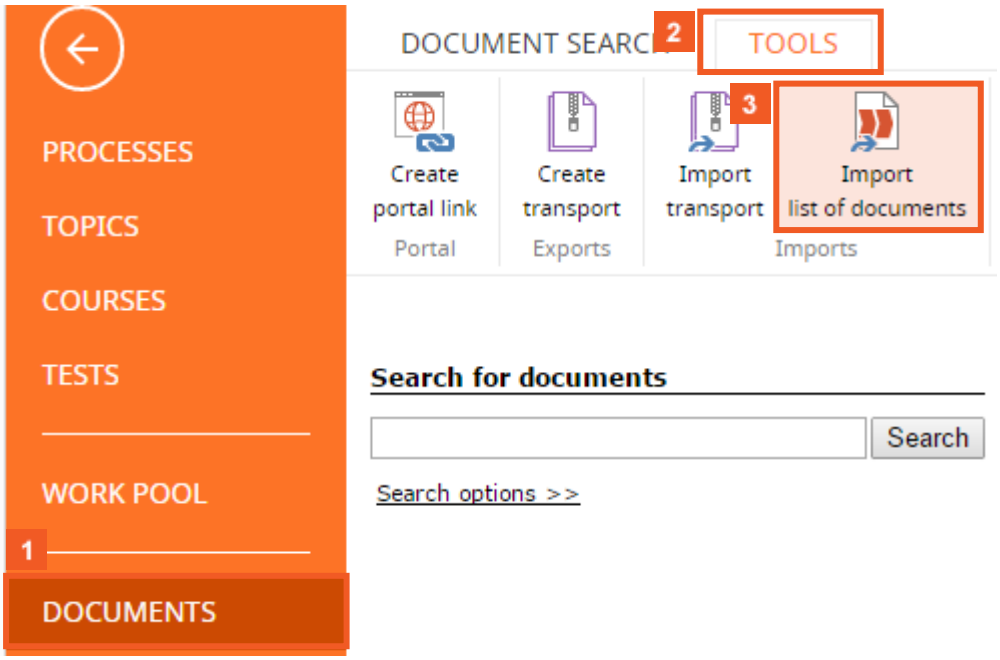
Please download the appropriate file here: https://docs.tts-cloud.com/publisher/search/?t=topic&t=document&q=excel+import+example

The "User import" ZIP file contains the following files:

- user_import.xls (Excel file for importing sample users)
- user_import - delete.xls (Excel file for deleting sample users)
- user_import.xml (Configuration file with the Import Descriptor for the sample import).

4 How to import documents along with their structures from Excel

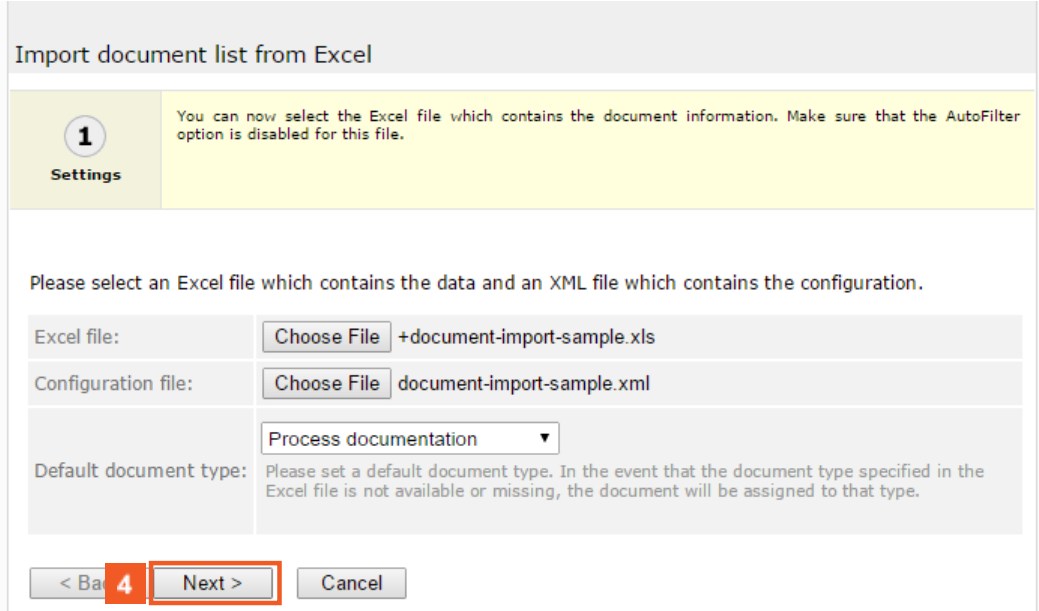
During our work on customer projects, we devised a function that facilitates the import of existing documents *en bloc*. This function – which has also been implemented as an administrator tool – is based on an Excel table that contains the attributes of documents, as well as the location where the associated document files have been stored.



1 Click the **Documents** tab.

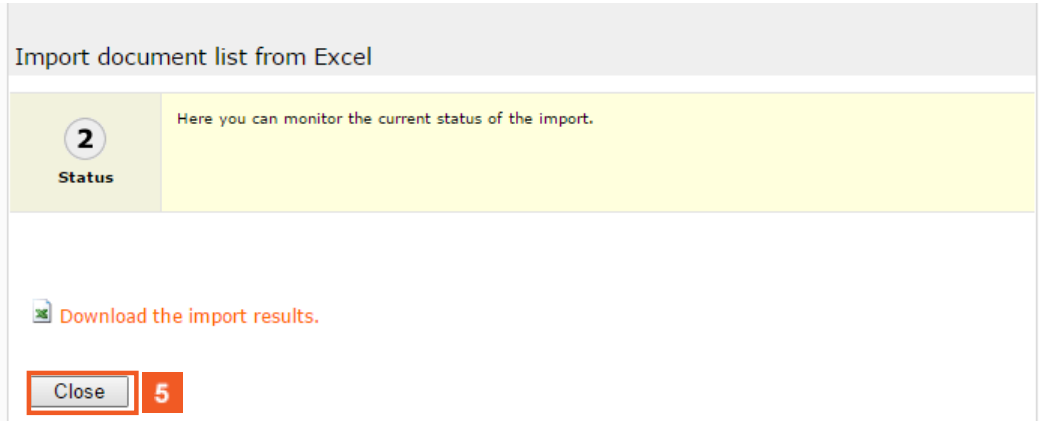
2 Click the **Tools** tab.

3 Click the **Import list of documents** button..



In the first step of the wizard, you have to select the appropriate **Excel file** (that contains the import description) and the corresponding **Configuration file** (XML).
You may also want to specify the **Default document type**, which will be assigned to all imported documents for which the type specified in the Excel file is not available.

4 Click the **Next** button.



In the second and last step of the wizard, you can choose to **Download the result** of the import process. It will be displayed as an Excel file.

5 To exit the dialog window: click the **Close** button.

5 Document import via Excel: Configuration

General information on the document Excel import

When performing a document Excel import, a standard document type has to be selected.

This will be used as the document type for the main dataset whenever the document type from the Excel file cannot be assigned.

Each row of the Excel file contains a main dataset (a document).

General parameters

Name	Value	Description
sheet	Examples: "sample" ("sample" sheet) "1" (1st sheet)	Name of the Excel sheet on which the import information is found, or the sheet no. (starting with 1)
importServerInstance	Technical name of the server instance for imported objects	Technical name of the server instance to which the imported objects are to be assigned. If it doesn't exist, a new server instance object will be created in the database.
dataBeginRow	Integer value beginning with 1	Indicates the row where the data begins. "2", for example, would mean that the data starts on the second row. The first row can then be used for column headers.

Examples from the configuration file:

```
<!-- first two rows may be used as the header -->
<set-parameter name="dataBeginRow" value="3"/>
<!-- name of the Excel sheet to be used -->
<set-parameter name="sheet" value="sample"/>

<!-- technical name of the server instance to be used/created -->
<set-parameter name="importServerInstance" value="importInstance"/>
```

Name	Value	Description
title	The document's title	The title has to be specified
language	The language of the document, e.g. de-de	The language has to be specified
resourceType	The document's resource/MIME type	The MIME type has to be specified, assuming you are not dealing with a placeholder document.
url	Relative URL within the file repository	The document's relative URL in the file repository. If a document with this URL already exists, only a placeholder will be imported (is merely used for checking – the actual document URL will be generated automatically).

repository	Technical name of the file repository	The file repository has to be assigned to the document type. If no file repository has been defined, the document will be created as a placeholder and will not have any content.
docType	Technical name of the document type	If no document type has been defined in the Excel file, the standard document type that was selected for the import will be used.
localPath	File path of the document that is to be imported	The local file system's path to the document that is to be imported. For example, (windows)"C:\x\y\file.txt" (unix) "/var/tmp/file.txt" If this attribute is not specified, the document will be imported as a placeholder and will not have any content.

The following attributes can be defined via an untyped extractor:

Parameter	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated for the technical name
property	The name of the attribute to be filled	Techn. name of the meta attribute (e.g. description, language etc.)

Untyped extractors receive only two parameters.

Example from the configuration file (untyped extractor for the language):

```
<extractor name="LanguageOfTheDocument">
    <set-parameter name="column" value="A"/>
    <set-parameter name="property" value="language"/>
</extractor>
```

	A	B	C	D	E	F	G
1	**Language of the document**	**Title of the document**	**Description of the document**	**Technical name of the document type**	**Technical name of the file repository**	**Local path to the document**	**MIME type of the document**
2	de-de	Test document import (txt)	This is the test document in plain text.	document	rep1	C:\...\plaintext_document.txt	text/plain
3	de-de	Test document import (ppt)	This is the test document as a PowerPoint file.	document	rep1	C:\...\powerpoint_document.ppt	application/vnd.ms-powerpoint
4	de-de	Test document import	This is the test document.	document	rep1	C:\...\tt5_document.ttcp	application/x-teamtraining-sequence

Example from the Excel file:

Parameter	Function
technicalName (optional)	Extracts the object's technical name (if no technical names have been specified, they will be generated automatically)
propertyValue (optional)	Extracts the value of a property that is assigned to the object (meta attribute)

Example from the configuration file (the document's technical name):

```
<extractor name="TechnicalNameOfTheDocument" type="technicalName">
    <set-parameter name="column" value="A/>
</extractor>
```

Example from the configuration file (the document's meta-attribute description – "description" is the technical name of the meta attribute):

```
<extractor name="DescriptionOfTheDocument" type="propertyValue">
    <set-parameter name="column" value="D"/>
    <set-parameter name="systemProperty" value="description"/>
</extractor>
```

The following typed extractors can be used during the document import.

Parameter	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated

Available parameters for the technicalName extractor:

Mode of operation:

As its input value, the **technicalName** extractor is given a column from which the document's technical name is to be read out.

Example from the configuration file:

```
<extractor name="name" type="technicalName">
    <set-parameter name="column" value="D"/>
</extractor>
```

Example from the Excel file:

	A	B	C	D
1	**Language of the document**	**Title of the document**	**Description of the document**	**Technical name of the document**
2	de-de	Test document import (placeholder)	This is the test document as a placeholder.	document01

Parameter	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated for the technical name
systemProperty	Technical name of the meta attribute	Techn. name of the meta attribute (e.g. description, language etc.)
partExpression	Regular expression	An optional regular expression. This describes the part of the meta attribute value that will actually be imported.

Available parameters for the propertyValue extractor:

Mode of operation:

As its input values, the **propertyValue** extractor is given the technical name of the meta attribute, as well as a column from which the value of the meta attribute is to be read-out.

Whenever you are dealing with a multilingual meta attribute, the meta-attribute value (assuming that the meta attribute for the object with the technical name exists) will always be created in the current editing language of the user carrying out the import.

Meta-attribute values of the meta-attribute lists type have to match the preconfigured ranges.

Example from the configuration file:

```
<extractor name="Description" type="propertyValue">
    <set-parameter name="column" value="C" />
    <set-parameter name="systemProperty" value="description"/>
</extractor>
```

Example from the Excel file:

	A	B	C	D
1	**Language of the document**	**Title of the document**	**Description of the document**	**Technical name of the document**
2	de-de	Test document import (placeholder)	This is the test document as a placeholder.	document01

Examples:

Sample files can be obtained from the tts docsportal.

Please download the appropriate file here: https://docs.tts-cloud.com/publisher/search/?t=topic&t=document&q=excel+import+example

The "Example: Document import..." ZIP file contains the following files:

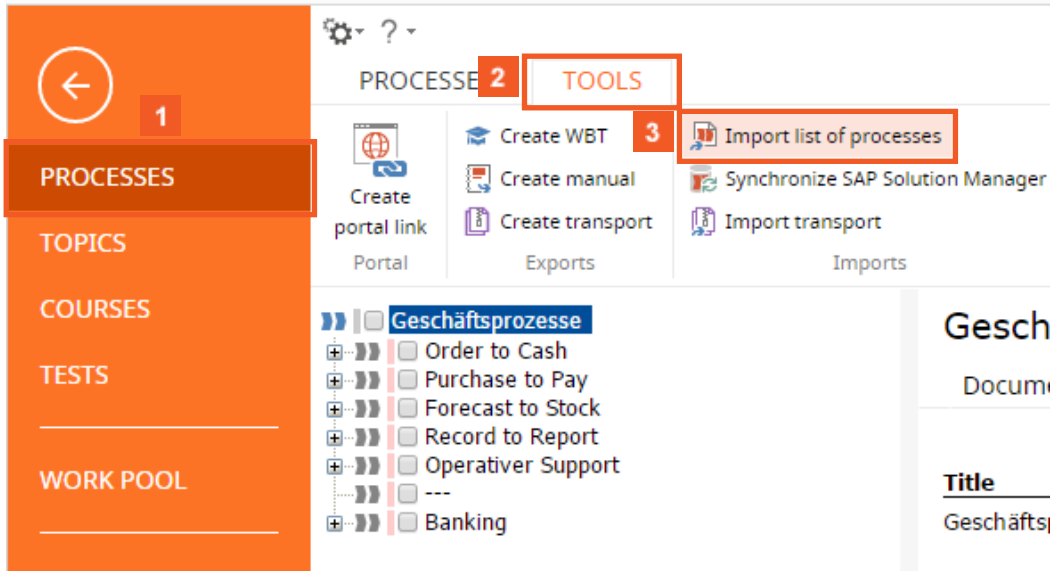
- document-import-sample.xls (Excel file for importing sample documents)
- document-import-sample.xml (configuration file with the Import Descriptor for the sample import)

Sample documents - with which the document import can be tested - are also included in a subfolder named "Dokumente".

6 How to import existing process structures from Excel

The Curator comes with a special function which facilitates the import of existing process structures from Excel tables. This means that you can skip the task of manually structuring a new process model.

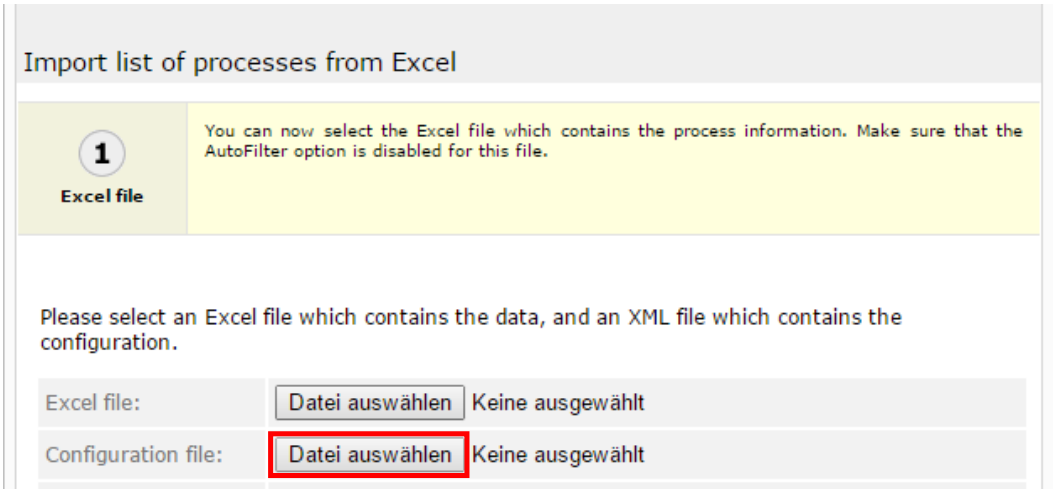
In addition to the process titles, any number of other process attributes can also be extracted – even the automatic recognition and assignment of roles is supported. The Process import function has been implemented as a tool and it can be accessed on the **Tools** tab in the **Processes** task.



1 Click the **Processes** tab.

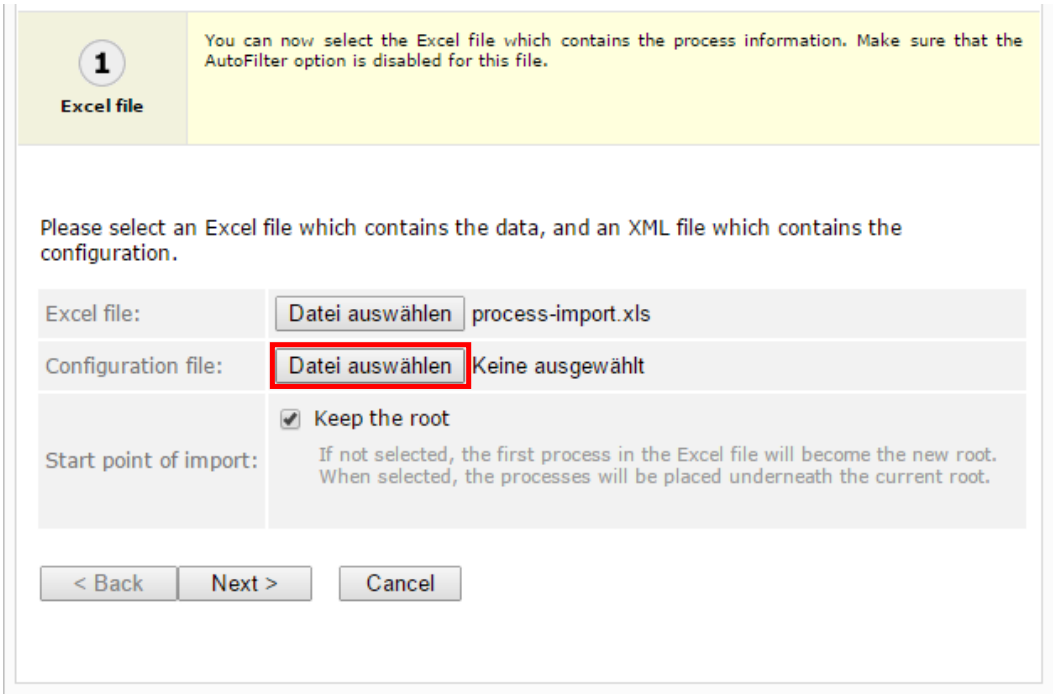
2 Click the **Tools** tab.

3 Click the **Import list of processes** button.



Select the Excel table that you want to use for the import from the file system.

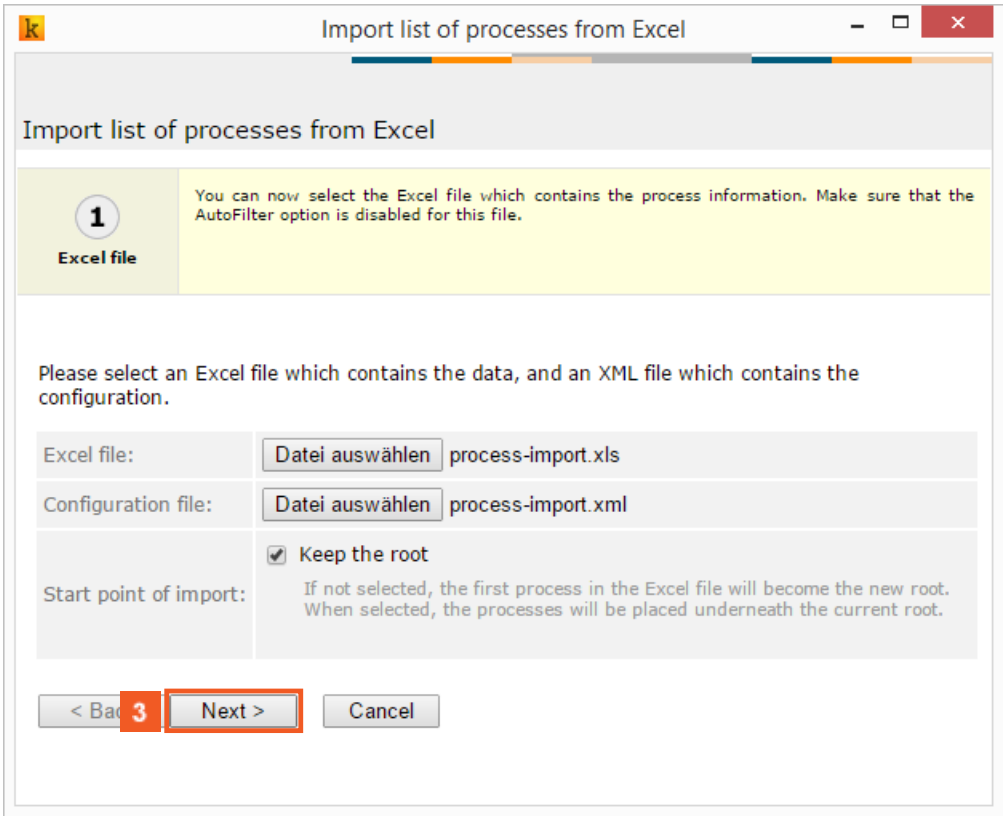
⚠ To ensure that the Excel file is imported without a hitch, it may not contain AutoFilters. Please remove any AutoFilters before importing.



In addition to selecting an Excel file, you should also choose an XML-based configuration file. The configuration file helps you to adapt the import mechanism to suit the actual structure of existing process tables. Among other things, it includes information on how the process information is distributed among the columns, which columns contain roles, etc.

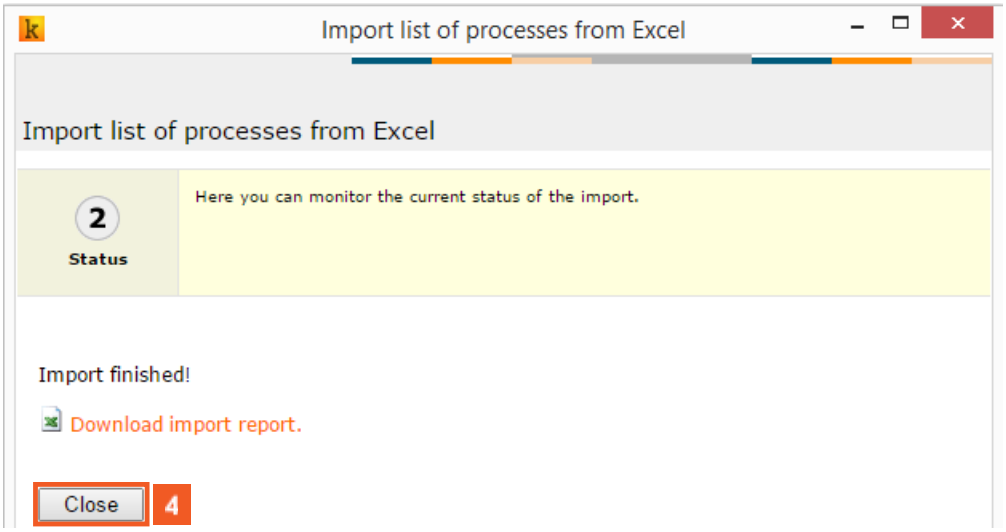
Use the appropriate **Browse** button to select the configuration file from your file system.

Due to the enormous functional scope and flexibility of the import mechanism, we will not discuss the actual structure of the configuration file here. Please refer to the next chapter for details.



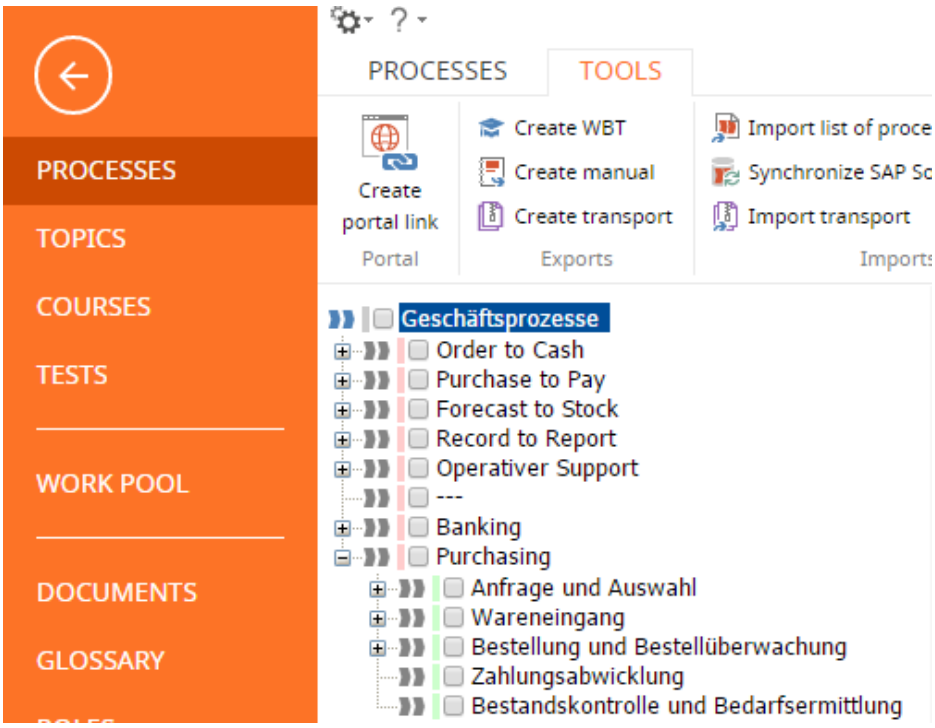
If you uncheck "Keep the root", you can force the current root process to be replaced by the first process in the Excel file.

3 Click the **Next** button.



If you wish to download and check the **result of the import** (as an Excel file), click the red hyperlink.

4 Click the **Close** button.



The processes have been imported.

7 Process import via Excel: Configuration

General information on the process Excel import

The process Excel import also supports the option to retain the root process. If this is selected, the root node will be replaced with the root from the import. However, this also means that all processes that used to be in the Curator will be lost.

Each row of the Excel file contains a main dataset (a process), as well as optional secondary datasets for roles / organizational units.

Both roles and organizational units can be created with this import. Roles are automatically assigned to the process (main dataset).

General parameters

Name	Value	Description
sheet	Examples: "sample" ("sample" sheet) "1" (1st sheet)	Name of the Excel sheet on which the import information is found, or the sheet no. (beginning with 1)
strategy	synchronizeProcesses	
importServerInstance	Technical name of the server instance for import-ed objects	Technical name of the server instance to which the imported objects are to be assigned. If it doesn't exist, a new server instance object will be created in the database.
dataBeginRow	Integer value beginning with 1	Indicates the row where the data begins. "2", for example, would mean that the data starts on the second row. The first row can then be used for column headers.

Examples from the configuration file:

```
<!-- first two rows may be used as the header -->
<set-parameter name="dataBeginRow" value="3"/>
<!-- name of the Excel sheet to be used -->
<set-parameter name="sheet" value="sample"/>

<!-- technical name of the server instance to be used/created -->
<set-parameter name="importServerInstance" value="importInstance"/>

<!-- always uses "synchronizeProcesses" -->

<set-parameter name="strategy" value="synchronizeProcesses"/>
```

Parameter	Function
technicalName (optional)	Extracts the technical name of the process (if no technical names have been specified, they will be generated automatically)
propertyValue (optional)	Extracts the value of a property that is assigned to the object (meta attribute)
entityValueList	Extracts technical names from a comma-separated list of the secondary objects (roles / organizational units) that are to be assigned.

The following typed extractors can be used during the process import.

Parameter	Function
processStructure	Extracts the position of the process within the Process tree
processLevel	Extracts the object's process level

The following process-specific extractors can be used.

Parameter	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated

Available parameters for the technicalName extractor:

Mode of operation:

As its input value, the **technicalName** extractor is given a column from which the technical name of the process is to be read out.

Example from the configuration file:

```
<extractor name="name" type="technicalName">
    <set-parameter name="column" value="E"/>
</extractor>
```

Example from the Excel file:

	A	B	C	D	E	F
1	Process title on level 1	Process title on level 2	Process title on level 3	Process level no.	Technical name of the process	Technical names of the assigned roles
2	Process 1 - Level 1			Global	11111	role1
3	Process 1 - Level 1	Process 2 - Level 2		Process	11112	role1,role2
4	Process 1 - Level 1	Process 2 - Level 2	Process 3 - Level 3	Process step	11113	role3

Parameter	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated for the meta-attribute value
systemProperty	Technical name of the meta attribute	Techn. name of the meta attribute (e.g. description, language etc.)
partExpression	Regular expression	An optional regular expression. This describes the part of the meta-attribute value that will actually be imported.

Available parameters for the propertyValue extractor:

Mode of operation:

As its input values, the **propertyValue** extractor is given the technical name of the meta attribute, as well as a column from which the value of the meta attribute is to be read out.

Whenever you are dealing with a multilingual meta attribute, the meta-attribute value (assuming that the meta attribute with the technical name exists) will always be created in the current editing language of the user carrying out the import.

Meta-attribute values of the meta-attribute lists type have to match the preconfigured ranges.

	A	B	C	D	E	F
1	Process title on level 1	Process title on level 2	Process title on level 3	Process level no.	Technical name of the process	Meta-attribute description
2	Process 1 - Level 1			Global	11111	A top-level process
3	Process 1 - Level 1	Process 2 - Level 2		Process	11112	A second-level process
4	Process 1 - Level 1	Process 2 - Level 2	Process 3 - Level 3	Process step	11113	A third-level process
5	Process 1 - Level 1	Process 2 - Level 2	Process 4 - Level 3	Process step	11114	Another third-level process

Example from the Excel file:

Example from the configuration file:

```
<extractor name="Description" type="propertyValue">
    <set-parameter name="column" value="F" />
    <set-parameter name="systemProperty" value="description" />
</extractor>
```

Parameter	Value range	Description
column	A - ZZ	Excel identifier for the column to be evaluated with the technical name of the process level
emptyStrategy	parentSuccessor	Adds the process level (assuming it's new) of the preceding process level. If no preceding process level exists, it will be added at the end.
value mapping		Specification regarding the conversion of the Excel identifier into the technical name of a process level.

Available parameters for the processLevel extractor:

Mode of operation:

The **processLevel** extractor reads out the specified column, converts the process level's identifier into its technical name and assigns the appropriate process level to the process.

The following fallback mechanisms are used for this:

- If a process level with this technical name exists, it will be used.
- If the superior process (parent node) has a process level which also has a subordinate process level, the subordinate process level will be assigned to the process.
- The same process level as that from the preceding imported process will be selected.

Value-mapping (optional)

Thanks to *value mapping*, the Excel file may contain *human-readable* identifiers for process levels. They will then be replaced with their technical names by the extractor. The specification regarding how they are to be converted can be found in the *value-mapping* element.

Parameter	Value range	Description
importValue	Arbitrary	Human-readable identifier for the process level
systemValue	Technical name of the process level	The tech. name of the process level

Example from the Excel file:

	A	B	C	D	E
1	Process title on level 1	Process title on level 2	Process title on level 3	Process level no.	Technical name of the process
2	Process 1 - Level 1			Global	11111
3	Process 1 - Level 1	Process 2 - Level 2		Process	11112
4	Process 1 - Level 1	Process 2 - Level 2	Process 3 - Level 3	Process step	11113
5	Process 1 - Level 1	Process 2 - Level 2	Process 4 - Level 3	Process step	11114

Example from the configuration file:

```
<extractor name="ProcessLevel" type="processLevel">
    <set-parameter name="column" value="D"/>
    <set-parameter name="emptyStrategy" value="parentSuccessor"/>
    <value-mapping>
        <mapping importValue="Global" systemValue="area"/>
        <mapping importValue="Process" systemValue="process"/>
        <mapping importValue="ProcessStep" sys-
temValue="processstep"/>
    </value-mapping>
</extractor>
```

Parameter	Value range	Description
startColumn	A - ZZ	Excel identifier for the first column to be evaluated
endColumn	A - ZZ	Excel identifier for the last column to be evaluated
strategy	fullpath	

Available parameters for the processStructure extractor:

Mode of operation:

The **processStructure** extractor is given a series of columns as its input values. Each of these columns in the Excel sheet contains a level from the Process tree. Put another way, one cell contains the title of the process in the corresponding level.
The processes (identified via their titles) from the lower levels are structured hierarchically according to the sequence quoted in the processes tree structure. Each dataset therefore contains the title of the process to be created, as well as its predecessors' titles in the Process tree.
The title of the process is always created in the current editing language of the user carrying out the import.

You need to ensure that the processes that are specified as predecessors either already exist or were imported before the actual process/dataset.

Example from the Excel file:

	A	B	C	D	E
1	Process title on level 1	Process title on level 2	Process title on level 3	Process level no.	Technical name of the process
2	Process 1 - Level 1			Global	11111
3	Process 1 - Level 1	Process 2 - Level 2		Process	11112
4	Process 1 - Level 1	Process 2 - Level 2	Process 3 - Level 3	Process step	11113
5	Process 1 - Level 1	Process 2 - Level 2	Process 4 - Level 3	Process step	11114

Example from the configuration file:

```
<extractor name="ProcessStructure" type="processStructure">
    <set-parameter name="strategy" value="fullpath"/>
    <set-parameter name="startColumn" value="A"/>
    <set-parameter name="endColumn" value="C"/>
</extractor>
```

Parameter	Value range	Description
column	A-ZZ	Excel identifier for the column to be evaluated, with the titles of the roles or organizational units. Titles are displayed in a comma-separated list here.
titleLanguage	Language code (RFC1766) e.g. de-de	To be used with the "column" parameter. This indicates the language to be used to search for a role / organizational unit with the specified title.
idColumn	A-ZZ	Excel identifier for the column to be evaluated in which the technical names of the role / organizational unit are found. Technical names are displayed in a comma-separated list here.
targetEntity	role, orgunit	Indicates that objects of the (process) roles or organizational units types are to be assigned. They will be created if they don't already exist.

Available parameters for the entityValueList extractor:

Mode of operation:

The **entityValueList** extractor supports two different ways of searching for existing objects.

- Via the technical names. The extractor is given a column - via the "idColumn" parameter - in which a comma-separated list of the technical names for roles / organizational units is expected.
- Via the title of the object. The extractor is given a comma-separated list of titles for roles / organizational units - via the "column" parameter - as well as the language, via the "titleLanguage" parameter. There is then a search for existing objects with these titles in the appropriate language.

If both exist, the search for existing objects always takes precedence over the technical names.

Roles and organizational units that do not already exist will be created.

Roles will be assigned to the process (main dataset), whereas organizational units will be merely created.

If a single row contains both an extractor for roles and an extractor for organizational units, this does not mean that the role is subordinate to the organizational units.

It is currently not possible to hierarchically subdivide imported roles or organizational units in the tree structure. All objects are created on the top level of the Role tree.

If technical names are indicated - but not the titles - the technical name will be used as the object title.

If the titles are indicated - but not the technical names - the technical name will be filled with the title (in a modified form if necessary).

Even when the "titleLanguage" has been indicated, the title will still always be created in the user's current editing language.

	A	B	C	D	E
1	Process title on level 1	Process title on level 2	Process title on level 3	Process level no.	Technical name of the process
2	Process 1 - Level 1			Global	11111
3	Process 1 - Level 1	Process 2 - Level 2		Process	11112
4	Process 1 - Level 1	Process 2 - Level 2	Process 3 - Level 3	Process step	11113
5	Process 1 - Level 1	Process 2 - Level 2	Process 4 - Level 3	Process step	11114

Example from the Excel file:

Example from the configuration file (assignment via the technical names of the process roles):

```
<extractor name="Roles" type="entityValueList">
    <set-parameter name="idColumn" value="F" />
    <set-parameter name="targetEntity" value="role" />
</extractor>
```

Important:

The order in which the extractors appear within the configuration file is important.

The following order should be used:

- technicalName
- processStructure
- processLevel
- entityValueList
- propertyValue

Examples:

Sample files can be obtained from the tts docsportal.

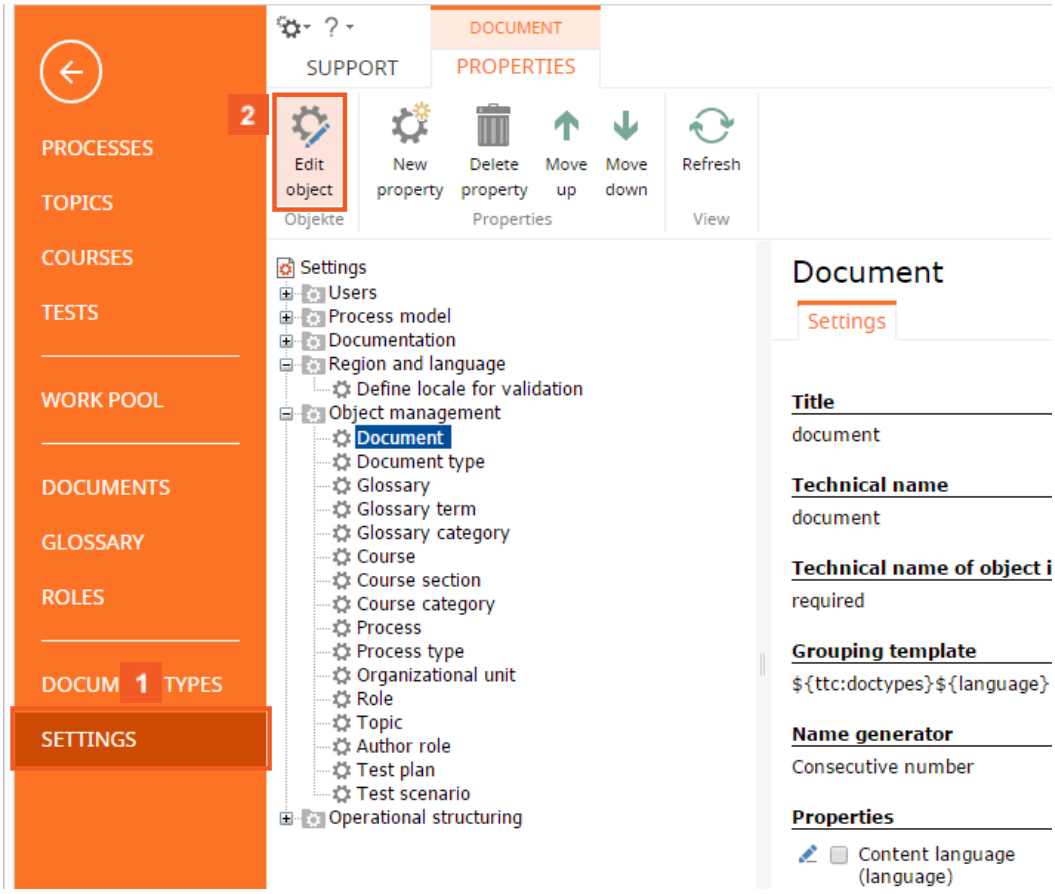
Please download the appropriate file here: https://docs.tts-cloud.com/publisher/search/?t=topic&t=document&q=excel+import+example

The "Example: Process import..." ZIP file contains the following files:

- process-import-sample.xls (Excel file for importing sample processes)
- process-import-sample.xml (configuration file with the Import Descriptor for the sample import).

8 Setting up which information is displayed in the list view

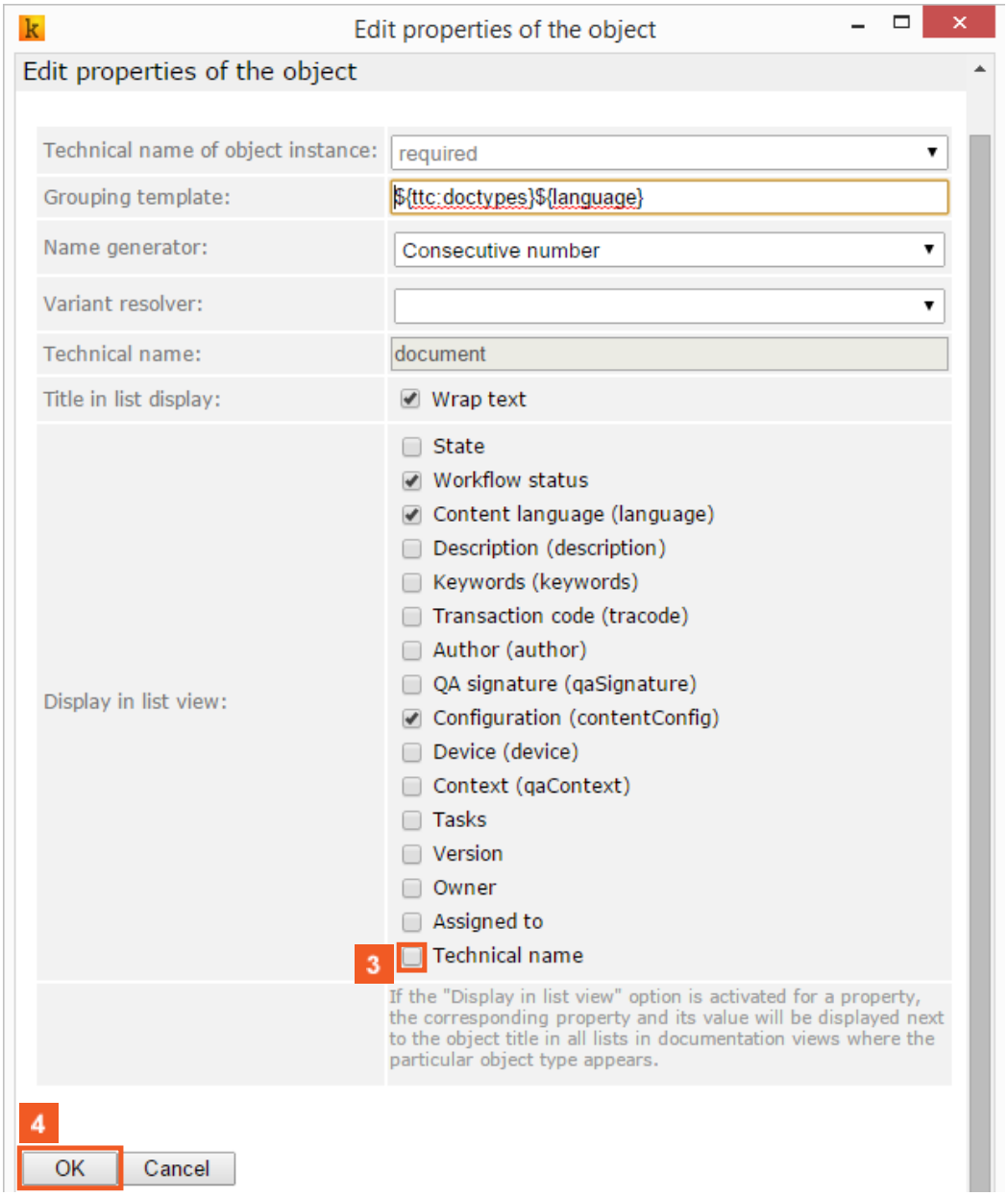
You can customize which data is displayed in the list view. In this example, the "Documents" list in the "Processes" view is to be modified to include the technical names of the listed documents. To accomplish this:



1 Click the **Settings** tab.

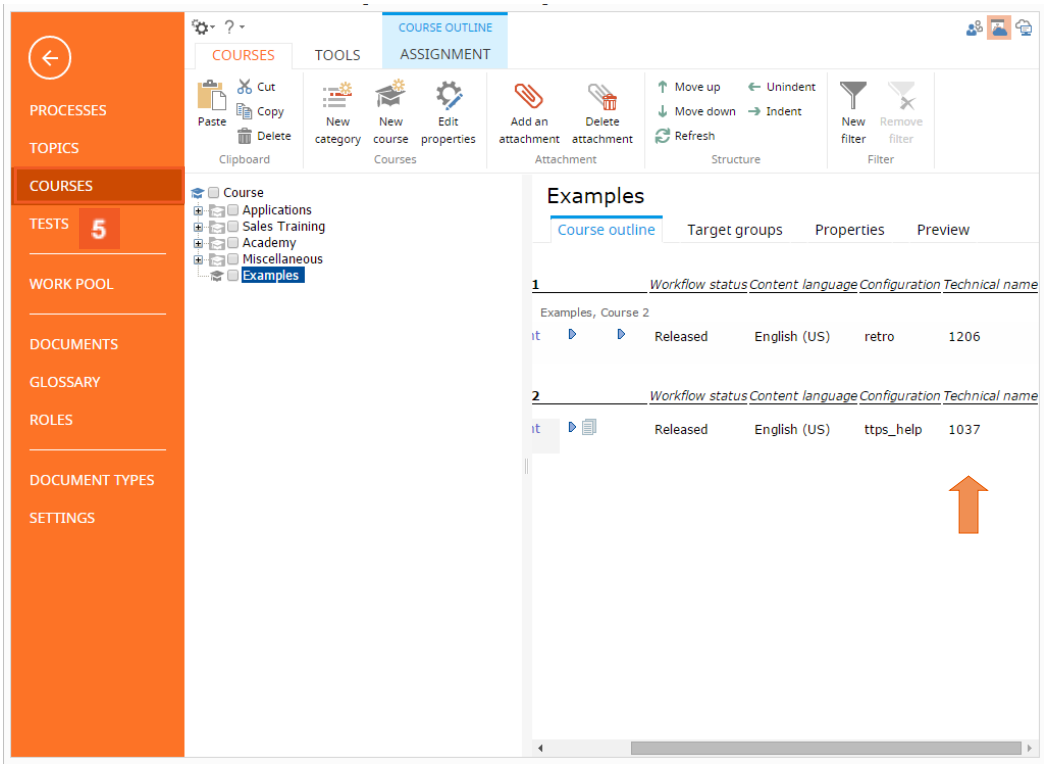
As the list settings for the "Document" object are to be changed, "Document" must be selected in the tree. As this is already highlighted, you can simply proceed now.

2 Click the **Edit object** button.



3 Click the **Technical name** checkbox.

4 Click the **OK** button.



5 Click the **Courses** tab.

As you can see, the list now contains the **Technical name** property which was just selected.