

enaio[®]

Software Documentation
enaio[®] editor

Version 8.50

All software products as well as all related extension programs and additional functions are registered and/or in-use trademarks of OPTIMAL SYSTEMS GmbH, Berlin or its subsidiaries. They may only be used according to a valid licensing agreement. The software as well as related documentation are protected by German and international copyright law. Unauthorized duplication and sales is plagiarism and subject to criminal prosecution. All rights reserved, including reproduction, transmission, translation, and storage with/on all kinds of media. For all preconfigured test scenarios or demo presentations: All company and person names which occur in examples (screenshots) are fictional. Any resemblance to existing companies or persons is purely coincidental and unintentional.

Copyright 1992 – 2017 by OPTIMAL SYSTEMS GmbH
Cicerostraße 26
D-10709 Berlin

07.04.2017
Version 8.50

Contents

Contents.....	III
Introduction.....	5
About the Manual.....	5
About enaio® editor.....	5
Versions and Configurations.....	5
Installation, Licensing, Security System.....	6
Overview of the Workspace.....	6
Defaults.....	9
Logging.....	10
Definition of Cabinets, Registers and Document Types.....	11
Object Definition - Introduction.....	11
Opening and Saving Object Definitions.....	11
Opening and Saving Active Object Definitions.....	12
Creating, Opening and Saving an Object Definition File.....	12
Importing an Object Definition.....	13
View and Print.....	14
Browse.....	14
Creating Cabinets, Register and Document Types.....	14
How to Create New Cabinets, Registers, and Document Types.....	14
How to Edit Cabinets, Registers, and Document Types.....	19
How to Delete Cabinets, Registers, and Document Types.....	20
How to Copy Cabinets, Registers, and Document Types.....	20
The E-Mail Document Type.....	21
Icons.....	21
Specifying Object Type Relations.....	23
Validating an Object Definition.....	24
The Performance Wizard.....	25
Defining Dialog Elements.....	27
Dialog Elements - Introduction.....	27
Creating Dialog Elements.....	27
Dialog Element Types.....	27
Creating Dialog Elements.....	29
Deleting Dialog Elements.....	30
Properties of Dialog Elements.....	31
Properties of Text Fields.....	31
Properties of Checkboxes.....	39
Properties of Radio Buttons.....	41
Properties of Group Fields.....	42
Properties of Tables.....	43
Properties of Page Controls.....	45
Properties of Web Controls.....	46
Properties of Images.....	47
Properties of Static Text.....	47
Properties of Buttons.....	48
Catalogs for text fields.....	48
List catalog.....	49

Tree catalog.....	53
Hierarchy catalog.....	54
Database catalog.....	55
Structure tree.....	56
Exporting and Importing Catalog Data.....	60
AddOns for Text Fields.....	60
Quickfinder AddOn.....	62
WWW AddOn.....	65
Date AddOn.....	65
Conversion AddOn.....	66
Filing Plan AddOn.....	66
Counter AddOn.....	69
ID AddOn.....	71
User AddOn.....	71
Rights group AddOn.....	72
Application add-on.....	74
Query AddOn.....	74
Address AddOn.....	75
Catalog AddOn.....	79
Database AddOn.....	87
VB Script AddOn.....	89
Multi-Fields.....	91
Layout of Forms.....	91
Selecting and Positioning Dialog Elements.....	92
Size of Dialog Elements.....	93
Aligning Dialog Elements.....	93
Copying, Inserting, and Deleting Dialog Elements.....	94
Tab Position of Dialog Elements.....	94
The Test Mode.....	95
Languages for Object Definitions.....	96
Multiple Languages – Introduction.....	96
Setting up Languages.....	96
Creation of Database Tables.....	99
Database Tables – Introduction.....	99
Checking Tables.....	100
Adjusting Tables.....	101
Removing Tables.....	102
Database indexing.....	102
Index.....	105

Introduction

About the Manual

This handbook is available as PDF file and as online help.

You can read the PDF file on the screen with Acrobat Reader, print it completely or in part, and browse it quickly for terms.

With the **Help** button or the menu, you can quickly open the online help tool on your desktop and search for the required topic.

All procedures described in the manual are based on mouse operation and use of the toolbar buttons. However, all operations can be performed with the keyboard as well. enaio® editor follows the conventions of MS Windows. Use **Alt** plus the underlined letters in the menu.

About enaio® editor

enaio® editor is a component of the enaio® content management, workflow and archiving system.

With enaio® editor, you can

- § create folders, registers and document types for enaio®,
- § create and edit the indexing forms of folder, register, and document types,
- § create tables in the database for folder, register, and document types.

Setting up and designing the folders, registers and document types has a large impact on work with enaio®. You can make it easier for users to capture and index documents and provide them with search methods which quickly and easily lead to the desired results.

The access rights for enaio® editor are defined in enaio® administrator. Users can receive the rights to start the enaio® editor and to edit catalogs, edit object definitions, and adjust the database. When a user starts enaio® editor, he must log in. Features for which you do not have access rights will be unavailable.

We recommend issuing enaio® editor access rights to only a few, very qualified users.

Versions and Configurations

As a rule, configurations are upwards compatible with different enaio® versions, that is, configurations created in earlier system versions can also be used in later system versions.

Downward compatibility is by contrast not guaranteed since enaio® platform developments result in new features and extended configurations. As a consequence, configurations created in later system versions cannot be used in earlier system versions.

It is very likely that attempting to import configurations that have been created in earlier system versions will result in feature settings being lost, format incompatibility, and/or program errors. Therefore you must pay attention to system versions when dealing with test, development and productive systems and in any case abstain from attempting to

import and activate configurations which have been created in later system versions. This applies to configuration files, in particular to object definitions and workflow models.

Installation, Licensing, Security System

enaio® editor is installed by the setup as part of the administration components. You can start it from the program group or from the folder `clients\admin`. There you can find it under the name `'axgredit.exe'`.

The 'ASE' module must be licensed for the workstation.

The user requires the system role 'Editor: Start'. You can then start enaio® editor, view and edit the current object definition. With the system role 'Adjust editor object definition' you can create and edit object definitions, with the system role 'Editor: Customize database' you can adapt the database after changes performed to the object definition.

If only the AXK module has been licensed, users with the system roles 'Start Editor', 'Editor: edit object definition' and 'Editor: Customize database' can import and customize databases.

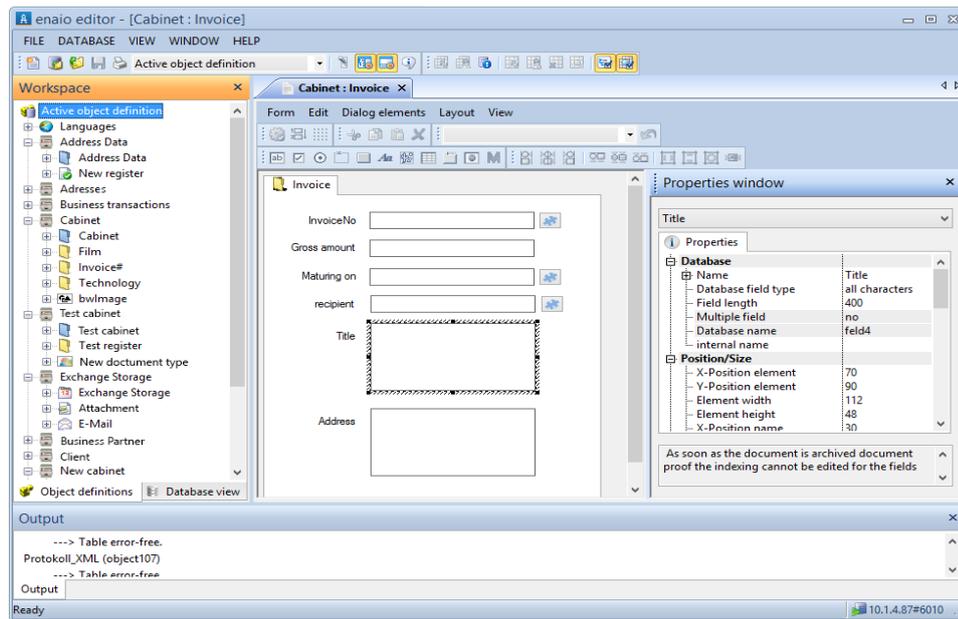
If only the 'AXK' module has been licensed, users who only dispose of the system role 'Start Editor' but not of the system roles 'Adjust Object Definition' and 'Editor: Customize database' can edit the catalogs. The database fit is not necessary when only catalogs are edited.

enaio® editor creates logs according to the settings from the configuration file `'oxrpt.cfg'`, located in the application directory `...\clients\admin`. Logging is documented in the Administration handbook.

Overview of the Workspace

Once you have started enaio® editor, in the program window you find:

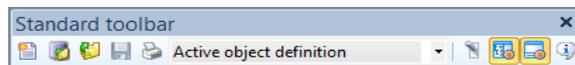
- § the menu bar,
- § the toolbar,
- § the workspace,
- § the workspace,
- § the output window,
- § and the status bar.



enaio® editor Program Window

The toolbar contains the following by default:

§ the administrative toolbar for object definitions,



§ the **Database** toolbar with the database features.



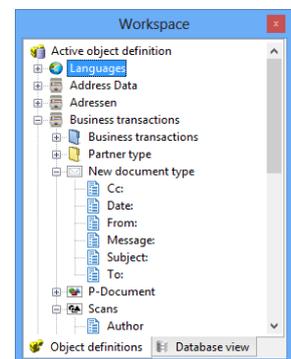
A new tab will be opened for every object you edit in the work area. The editing features can be accessed via the menu bar, the toolbars, and the context menu. Toolbars can be shown/hidden, moved anywhere with the mouse, or docked to the window edge.

The workspace is divided into two tabs.

On the **Object definitions** tab, the archive objects, folders, and register and document types and their fields are listed in a tree structure.

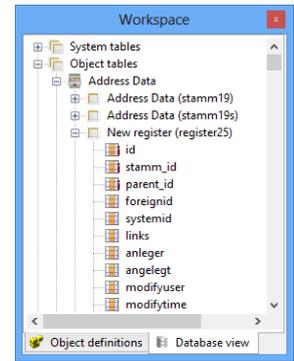
You can select archive objects and open the indexing forms for editing in the work area.

Moreover, you can add and delete languages to be used. For each language listed here, own names can be specified.



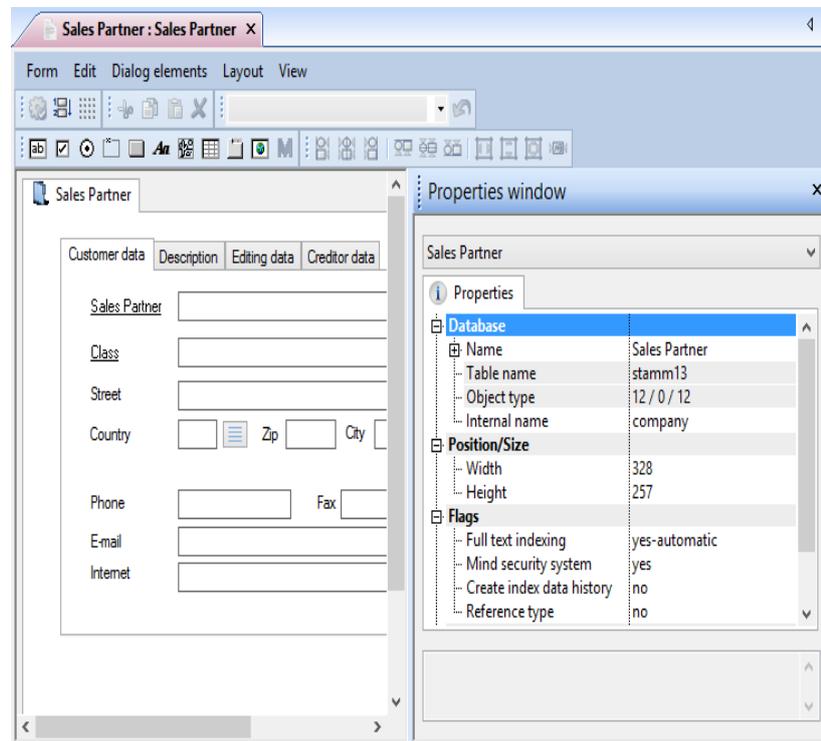
On the **Database** tab, the database tables are listed in a tree structure.

You can select, check, adjust, or delete tables.



You can also show/hide the workspace, move it to a different position or dock it. When dragging the area with the mouse, a positioning tool appears. You can drop the area to any position on the workspace or dock it by using the positioning tool.

The indexing forms of the archive objects which you are editing are displayed in windows on the work area. Archive objects and the fields of the indexing forms contain properties which you can determine in the properties window. In the properties window the properties for the selected objects are displayed.



Using the **Windows** menu, you can sort the windows and switch between them.

You find the output window displayed by default at the bottom of the program window. All database features are logged in this window.



The bars and windows can be opened and closed via the menu **View** or via the context menu of the work area. The settings related to the workplace are saved.

If you hold down the **F8** key while starting enaio® editor, all window settings will be replaced with the default settings.

Defaults

enaio® editor is initially started with the default settings which you can edit via the **File / Settings** menu.

The settings dialog is divided into six tabs.

Tab 'maximum database field length'

Here you specify the maximum number of characters which your database allows for a text field. enaio® editor automatically corrects new entries for the field length of dialog elements which have exceeded the specified maximum database length.

After making changes you can verify the settings for the object definition using validation (see 'Validating an Object Definition').

Tab 'Default grid settings'

Here you can define default grid settings. For every form you may choose a different grid setting (see 'Layout of Forms').

Tab 'min / max form size'

New archive objects are created with the standard size. The maximum dialog size cannot be exceeded and the minimum size cannot be violated.

After making changes you can verify the settings for the object definition using validation (see 'Validating an Object Definition').

Tab 'Miscellaneous'

The active object definitions from the database can be loaded automatically when starting enaio® editor.

You can select if the identification of dialog elements should be shown in the form via the property 'Label not visible.' In the test mode it will always be hidden.

Forms with more than 240 dialog elements from older versions can cause problems. They can only be edited if the option 'Form with more than 240 dialog elements is fully editable' is selected. These dialogs can only be edited in order to reduce the number of dialog elements or to divide the dialog elements via the dialog element 'Page control'.

If you select the feature **Automatically save window layout of form window on closing**, the last size setting will automatically be saved.

Tab 'Confirm'

You can switch on / off a range of confirmation checks, especially for deletions.

Tab 'XSL selection'

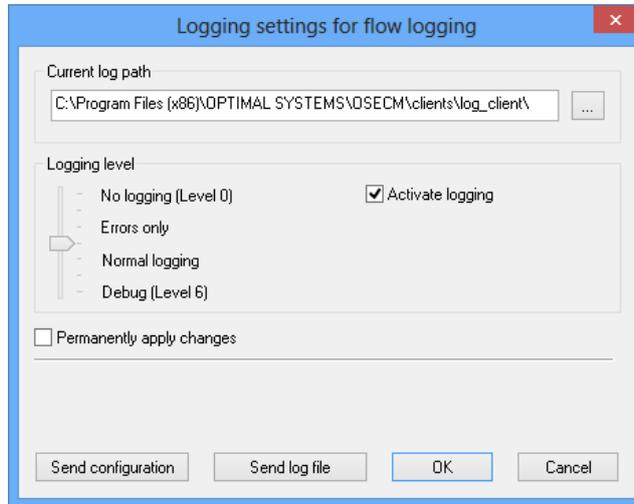
Here you can define a default XSL for formatting the object definition file or you can specify that an XSL must always be selected. The XSL formats the object definition for printing and for the table view.

Tab 'Output logging'

The results of the table check and fitting will be displayed in the output window. Additionally you can specify a file in which the results should be written.

Logging

In enaio® editor, the logging depth of the 'Default' channel can be set up for the duration of the session. The change of the logging depth will immediately apply; enaio® editor does not have to be reloaded. In case of error, it enables you to instantly send the relevant information to the administrator.



You can open the dialog via the menu **File / Log settings**.

The **Send configuration** button is used to send the currently set logging configuration `oxrpt.cfg` by e-mail. In doing so, all log files (*.evn) of the current day and the configuration file will be send as a ZIP archive.

Users with the system role 'Administrator: Configure entire system' can permanently adopt the logging level settings.

Definition of Cabinets, Registers and Document Types

Object Definition - Introduction

The object definitions of the folder, register, and document types for the content management, workflow, and archive systems of enaio® are administered in the database.

During the installation of enaio® you can specify a file containing the object definition. This object definition can then be imported.

You can also start enaio® editor without preset object definitions and then create new object definitions. Here you can import object definitions that are in file form, or open them and just adopt parts and add them to the active object definitions.

With enaio® editor you can also create and administer object definitions in files.

When you edit the active object definitions you usually have to adjust the database with enaio® editor afterwards. If you create new folder, register, or document types, they will only be available to users after the database has been updated, and either the enaio® server has been restarted or the engines have been reloaded. Users must also be given specific access rights in enaio® administrator, and each user's enaio® client must be restarted.

Before you reload the engines or update the database tables, you should validate the object definition (see 'Validating an Object Definition')

Opening and Saving Object Definitions

enaio® editor administers active object definitions. The database has to be updated with the active object definition so that users can create and search for folders, registers and documents.

Additionally, object definition files can be edited. From the object definition files, single object definitions can be adopted for the active object definition. Also, object definition files can be imported, so that active object definitions can be replaced by object definitions from the object definition file.

Object definition files are administered in the XML format. The data for structure tree catalogs (see 'Structure tree') are administered in their own files that can be named as you choose. These files must be available if you administer the object definitions with structure trees in a file format.

Each object definition is 'active'; this means that via the button **Save** on the standard toolbar or via the entry **Save object definition** from the **File** menu it can be saved quickly.

The object definition which has last been edited is always 'Active.' The active object definition is displayed on the standard toolbar.

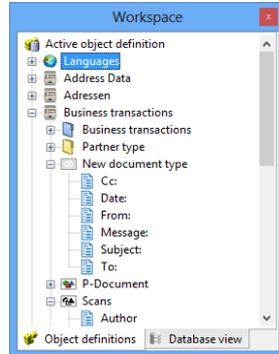
Opening and Saving Active Object Definitions



With the button on the standard toolbar or the **Open current system object definition** entry from the menu **File** you can open the active object definition.

They will be displayed as a tree structure in the Workspace.

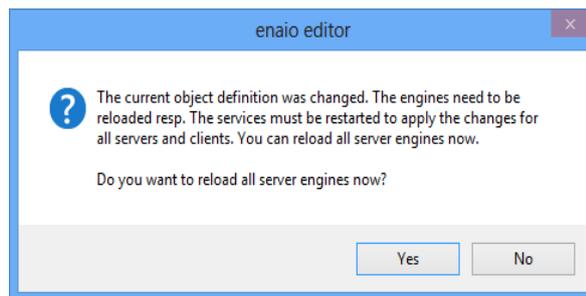
The active object definition will be displayed in the standard toolbar as a list entry.



With the button on the standard toolbar or the entry **Save object definition** from the **File** menu, you can save the active object definitions while they are active.

Via the context menu in the workspace you can directly save a selected object definition. You can save the active object definition via the entry **Save object definition** from the context menu as an object definition file.

After changing the active object definitions, enaio® server has to be restarted or the engines have to be reloaded so that enaio® server has the updated data. When you save the object definition you are presented with the option to reload the effected engines.



You can switch off this notification by changing the default settings on the tab **Confirm** to specify that the server engines should automatically be reloaded.

Via the entry **Reload server engines** in the **File** menu you can also let the engines be reloaded and in this way enable changes.

In multiple server/server group environments, all server engines are reloaded, but the object definition file `asobjdef.xml`, located in the directory `\server\etc\`, is not automatically copied to the corresponding directory in all other server groups.

After changes to the current object definition which require the table to be adapted a notification will be displayed which allows you to adapt the tables instantly.

Creating, Opening and Saving an Object Definition File



With the button on the standard toolbar or the entry **Create new object definition file...** from the menu **File** you can create a new object definition file.

In the file selection dialog select a folder and enter a name for the file.



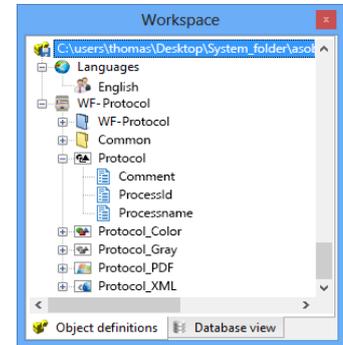
With the button on the standard toolbar or the entry **Open object definition...** from the menu **File** you can open an object definition file.

You can select a file using the file selection dialog.

The data for structure tree catalogs (see 'Structure tree') are administered in their own files. These files have to be available if structure trees were created.

The new or opened object definition file is displayed in the workspace.

In contrast to the active object definitions, the object definition files in the workspace are indicated by a floppy disk icon.



Via the entry **Save object definition** from the context menu of the object definition selected you can save the object definition. Via the entry **Save object definition as** you can also save the object definition file under a different name or in 'HTML' format.

If the object definition is 'active', it can be saved with the button **Save object definition** on the standard toolbar or via the entry **Save object definition** in the **File** menu.

Importing an Object Definition

When importing an object definition file the active object definitions are replaced by the object definitions from the object definition file, including the object type relations.

Select the entry **Import object definition** from the **Database** menu.

A confirmation dialog will be displayed.

Please note that an import cannot be undone. For safety reasons, always save the old object definition as a file before importing. Via the confirmation dialog you can delete the access rights for the archive objects, which were assigned in the enaio® administrator, at the same time.

You can select a file using the file selection dialog.

The imported object definitions appear as the active ones in the workspace. They only have to be saved if you edit the object definitions.

After importing an object definition the database has to be adjusted.

Make sure that structure trees are administered using their own files and that they are accessible. Check whether internal names contain special characters which are not valid in the active version.

If you have opened the object definition file, you can import this file into the database via the entry **Import object definition** from the context menu of the object definition file in the workspace.

View and Print

Opened object definitions can be displayed in the table view and printed.

These features are accessed via the **File** menu.

An XML version is displayed in a table view by formatting it with a style sheet. The style sheet is also used for printing and for saving in 'HTML' format. Specify a default style sheet in the settings and select it.

If you select a cabinet in the workspace, only the folder, register and document type data from that cabinet will be displayed or printed.

Browse

You can browse opened object definitions. A dialog will be displayed with columns for the field name, the internal name, and the database name.

These functions can be accessed via the **File** menu or the keyboard shortcut **CTRL+F**. You can browse individual cabinets and object types with the context menu in the workspace.

If you enter characters into the search field of a dialog, only entries containing these characters will be displayed. Double-clicking an entry will open the object type.

Creating Cabinets, Register and Document Types

The archive consists of cabinets which are identified by a folder type, and register and document types which are set up for the cabinet. Registers and documents are always created in folders by users. Via the object relations you can determine which level of the cabinet the register and document types should be created on.

You can create, edit and delete folder, register and document types.

Definitions of folder, register and document types can be copied within an object definition or copied to a different object definition.

You can create language versions for the names of the archive objects, dialog elements, and for the tooltips (see 'Multiple Languages – Introduction').

Folder, register and document types are edited in the Workspace.

If you create new objects, the database has to be updated afterwards. If you delete objects you must at the latest update the database before you create new objects.

New archive objects are only available in enaio® client after restarting enaio® server or reloading the engines.

How to Create New Cabinets, Registers, and Document Types

When creating an archive, you first create the cabinet (which is defined in terms of a folder type), then you specify the folder types and the document types which the users can use in this cabinet.

For naming the cabinets, register, and document types you can use a maximum of 30 characters, including special characters. The character combination ']]>' is not permitted to be used for names. If you use a semicolon you cannot include the address AddOn.

The size of the dialog is specified in dialog units when they are created. One dialog unit is equivalent to around about 1.5 screen points. For a screen resolution of 1024 x 768 we propose a maximum form size of 500 x 350 dialog units. Dialogs of this size will be displayed without a scroll bar in the standard view in the workspace.

Via Settings you can define a minimum and maximum dialog size.

Folder, register, and document types can be given internal names. A maximum of 100 characters is allowed for an internal name. The internal names for the object types should be obvious, and in an object definition the internal name of the dialog elements has to be obvious. The internal name is preset with the assigned name by default, and invalid characters are replaced with '_ '.

For the first character of an internal name the characters 'a'-'z', 'A'-'Z' and the character '_ ' are allowed. For the other characters numbers and the dot are also allowed.

Internal names are often used if you work with AddOns which refer to other dialog elements, or when additional components access the object definitions.

Internal names for folder, register, and document types are also required if users set different languages for the object definition. If e.g. a user sends an internal reference for an object, the recipient can only open this object in enaio® client if he uses the same language or if an internal name was specified for the object type name.

Cabinets and Folder Types

A maximum of 255 cabinets can be integrated.

These are the steps to create a new cabinet and folder type:

1. Open an object definition or create a new object definition.
2. Select the object definition in the workspace.
3. From the context menu select **Add cabinet**.
4. Enter a name, and optionally an internal name, in the dialog.
5. Confirm with **OK**.



The new cabinet is displayed in the Workspace. It has a folder assigned which has the same name.

New archive objects are flagged with a '*' and a '!' in the workspace. The '*' marks objects whose files are either not or only incompletely saved in the object definition; the '!' marks objects whose tables have not been set up or whose tables have to be modified after changes.

By double-clicking the folder type you open the new folder indexing dialog and the properties window on the desktop.

Opened archive objects are flagged with a '#' in the workspace.

The properties which are in the gray rows are automatically issued and cannot be edited. The rows which are white can be edited in the second column. You enter values or select them from the list.

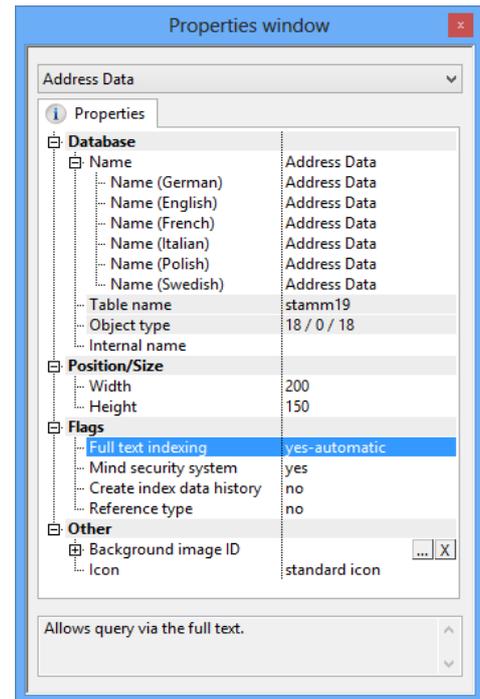
The folder types always have the Identifier of the cabinet name. Table names and object types are automatically administered.

The entries for the Object type contain 'Object type / Main type / Sub type'. This data and the Table name are partially necessary for configuring additional components.

A standard size is used as the default for the Width, Height, and Size of the folder indexing dialogs. These entries can be edited or you can resize the indexing dialog to the intended size with the mouse on the desktop.

For folder types you can set up full text indexing. The user receives an additional query feature: querying of the full text index. When using the automatic full text indexing the data will be full text indexed after creation and after every editing. When using non-automatic full text indexing the data is provided for the full text indexing by an automated action.

Additional information is found in the 'Full Text Indexing' manual.



Some additional components of enaio® can transfer data from MS Word documents, or access archive data and other documents via COM functions for example. In this case it can make sense to allow access which ignores the access rights of the users. If you select **Enforce security system: no**, you allow access of COM features independently from the access rights which were assigned in enaio® administrator. The access rights for the users in enaio® client are not affected.

For every object the client files an editing history. Here all changes in the object are logged. If you create an index data history, the old indexing versions are saved and can be displayed and restored in the editing history. The index data history settings can be defined by the user. The index data history settings can be defined by the user. Users with the system role 'enaio® client: Configure history for single objects' can switch on / off the index data history for each object of a given type via the object information.

If you assign the property Reference type-Yes to an object type, objects of this type can be used as reference objects for relations.

Folder types, as well as register and document types can have an icon assigned (see 'Icons').

Register Types

Register types are assigned to a cabinet.

Follow these steps to create a register type:

1. In the workspace select a  cabinet from an object definition.
2. Select **Add / Register** from the context menu (menu key).
3. Enter a name, and optionally an internal name, in the dialog.
4. Confirm with **OK**.



The new register is displayed in the Workspace.

By double-clicking on the  register type you open the indexing dialog and property window for the new register in the desktop.

Unsaved archive objects are shown in the workspace with a '*' and opened archive objects are shown with a '#'. Objects with tables which required adjustment are flagged with a '|'.

Register types and folders have the same editable properties: Width, Height, Full text indexing, Enforce security system, Create index data history, Reference type, Background image, and Icon.

Document Types

Document types are also assigned to cabinets. Every document type has a module assigned which enables creation and display of documents of this type.

Follow these steps to create a document type:

1. In the workspace select a  cabinet from an object definition.
2. Select **Add / Module type** from the context menu (menu key).
3. Enter a name, and optionally an internal name, in the dialog.
4. Confirm with **OK**.

The following modules are available:

 The module for grayscale images. Grayscale images are saved in the JPEG format.

 The module for black-and-white images. Black-and-white images are saved in the TIFF G4 format.

 The module for color images. Color images are saved in the JPEG format.

 The module for Windows documents. W-Document types are assigned an application in enaio® administrator. W-Documents are edited in this assigned application and saved in the corresponding format.

 The module for videos. Users import videos as 'MPEG' or 'AVI' files.

 The module for e-mail. E-mail messages from a MAPI 1.0 compatible mail system can be administered in enaio®. The fields in this document type require special identification (see 'The E-Mail Document Type').

 The module for XML documents. Style sheets are assigned to XML document types by adding entries to the file `as.cfg`.

You can find information about this in the 'enaio® Administration' handbook.

 The module for container documents. Users can compile any files in a container and check them in/out, edit and match in the directory structure.

Thus, the files are managed in a ZIP archive.

Files can be no more than 2 GB in size, and videos no more than 4 GB. For image modules, documents can consist of multiple files, and the sum of all image files cannot exceed 4 GB. The maximum size is limited to 1 GB by default. This setting can be changed in enaio® enterprise-manager via **Server properties > Category General > Maximum file size**.

New document types are displayed in the workspace with their corresponding module icon. You can assign different icons to every document, folder, and register type (see 'Icons').

For document types which you want use across multiple modules, you first specify a single module. Via the properties window you can then assign the property 'module-spanning document type'. For these documents the user selects a module after indexing. In the hit lists and in the folder window the module icon of the document type will be displayed, and in the archive section a standard icon is used for the document type: .

If you select a W-Document as module-spanning document type, variant management can be used.

The e-mail module must be enabled by adding an entry to the `as.cfg` configuration file, located in the `\etc` directory of the data directory, and the XML module can be disabled in the same way:

```
[SYSTEM]
EOBJECTMENUMODE=0
XMLOBJECTMENUMODE=1
```

Only licensed modules are available for the user.

By double-clicking the document type, you can open the properties window and the editing window with the document indexing dialog in the work area.

The editable properties: Width, Height, Full text indexing, Enforce security system, Reference type, Background image, and Icon correspond to the folder and register properties.

In the Height (Multiple fields) row enter the height for the section in the dialog, where multiple fields can be used. This section is shown on the bottom of the dialog. The width is automatically adjusted to the width of the form (see 'Multi-Fields').

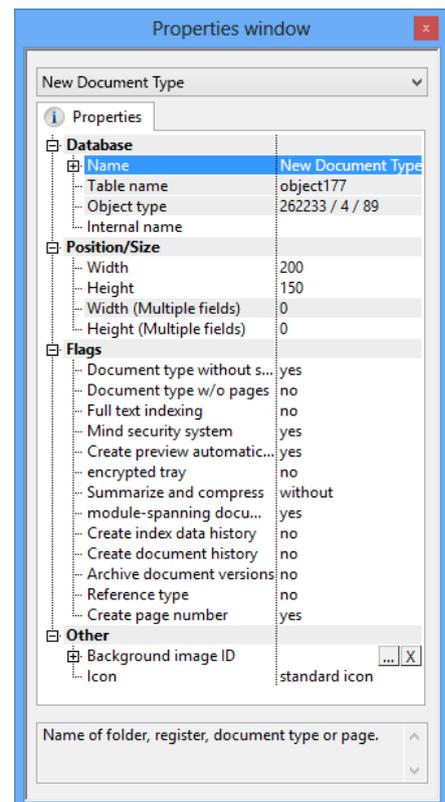
No quicklooks (minimized previews that improve clarity) are created for a document type without slides. Instead of Quicklooks users will see a default icon in enaio® client.

When creating quicklooks for image documents whose width or height is greater than 65535 pixels, errors may occur.

A document type without pages consists of the indexing only. No document file is assigned to it.

Document types without pages are displayed with a default icon in the client: .

Users can create documents of any document type in enaio® client in such a way that a document file is assigned at a later point or is not assigned at all.



Previews can be automatically created for the documents. These content previews can be displayed in the DocumentViewer. The page number can also be generated for documents. The page number is shown in the object information.

Documents can be encrypted for storage and transport by enaio® client. Therefore use the encryption module (see Administrator handbook). All document types with the property Encrypted filing will be included.

Document types with the property Encrypted filing must be created as document types without slides.

enaio® server can encrypt documents regardless of the Encrypted filing property.

No full text index can be created for documents encrypted by enaio® client.

The property Combine and compress determines whether document files should be individually combined and additionally compressed when they are archived. Combining makes sense for grayscale, black-and-white, and color image document types which contain multiple images. The single image files are combined in a CAB file. This way they require less storage capacity. Only document files containing up to 2000 image files can be archived with the property Combine and compress set.

Compression can also make sense for W-Documents. Compressing a CAB file can also save storage capacity.

For every object the client files an editing history. Here all changes in the object are logged. For document types you can set up a document history and an index data history.

In the index data history the old versions of the indexing data are saved after changes have been made to the indexing. In the document history the old versions of the document itself are saved.

Via the editing history, old document versions and indexing versions can be displayed and recovered.

Histories can be set up by the user. The index data history settings can be defined by the user. Users with the system role 'enaio® client: Configure history for single objects' can switch on / off the index data history for every object via the object information.

If you select Archive document versions, all old versions of the documents are archived in addition to the current version.

Annotations on layers for a document are not included in the document versions. By editing history documents are always displayed with levels of the current version. If you use document history for W-Documents the version of the document is always saved when checking in the document, even if no changes have been made in the W application.

Module-spanning document types are document types where the name of the document type as well as index and search forms are already given, but the module for creating a document can be selected by the user.

If you have selected the document history for module-spanning document type, you must note the file format for the document type when you import data. If a document has versions with file formats which are assigned to different modules, opening and recovering the old versions via the user administration will lead to errors.

How to Edit Cabinets, Registers, and Document Types

You edit the properties of cabinets, registers and document types via the properties window which you open by double-clicking the archive object in the workspace.

In the properties window the properties are displayed for the object which is selected in the workspace. The editable properties have a white background.

If you edit properties they will be selected with a '*' in the title row, the dialog and the property window until the changes have been adopted. A '!' is inserted in the workspace when changes are required by the adjustment of the table.

The editable folder, register, and document type properties can be edited without having to update the database.

For example if you change a document type with pages to a document type without pages, users can no longer display the existing pages.

How to Delete Cabinets, Registers, and Document Types

Cabinets, register and document types are deleted via the workspace. In the context menu of the register or document type you find **Delete object type**, in the context menu of a cabinet **Delete cabinet**. You will be prompted to confirm the action.

With the key **Del** you can also delete the selected object type or cabinet.

Save the object definition afterwards. If you have not yet saved the object definition the object will be marked with a red cross in the workspace. Via the context menu you can undo the deletion.

If data have already been captured for an object type, you will be prompted to confirm the action and can specify whether the associated database tables are deleted along with the indexing. If you select 'No,' the database tables will not be deleted, and will be listed under unused tables in the database view. If you cancel this operation nothing will be deleted.

Please note that there can be cross references between objects and when you delete one object the reference will lead to a blank spot. Edit the properties of any linked objects before deleting an object. If the deletion has an effect on the object relation you will receive a warning before you delete it.

Unused tables must be deleted before creating new cabinets, register or document types.

How to Copy Cabinets, Registers, and Document Types

Cabinets and register and document types can be copied within an object definition or to a different object definition.

§ Cabinets are dragged to an object definition by holding the left mouse button.

Cabinets are copied with their contents.

§ Register or document types are dragged to the  cabinet of an object definition by holding the left mouse button.

If an object with the same name or internal name already exists you will be asked to rename the copied object.

Please check if you have to adjust the dialog elements with catalogs, AddOns, or icons. Please also check if the internal name contains special characters which are not valid anymore in the active version.

The E-Mail Document Type

If you are managing e-mail messages in enaio®, set up an e-mail document type for every cabinet which should contain e-mail messages.

E-mail document types are automatically set up with the following fields:

Field	Internal name
From:	MAIL_FROM
To:	MAIL_TO
Cc:	MAIL_CC
Subject:	MAIL_SUBJECT
Date:	MAIL_SUBMIT_TIME
Message:	MAIL_BODY

When transferring an e-mail from the inbox to enaio® client, the client automatically assigns the identification data of the e-mail to the index data fields via the internal name. When inserting an e-mail, the fields are automatically indexed with the corresponding data.

Additionally you can set up an invisible field for the addressee of a blind copy with the internal name 'MAIL_BCC'. This field will also be automatically indexed with data from the e-mail.

The 'Date' field has the database field type 'Date/Time.'

You can edit the dialog elements in any way, for example you can change the name, the field length, the length of the database fields or you can delete fields which you do not need anymore.

The 'Message:' field has a default length of 248 characters. If the database allows more characters you should extend the permitted number of characters.

A document type can have any number of dialog elements added.

Details on creating dialog elements can be found in the section 'Creating Dialog Elements.'

If the text of an e-mail which is inserted to the field 'Message' is longer than the maximum allowed length of the text field, the exceeding text will not be transferred to the data sheet in the 'Message' index data field, even though it shows in the e-mail module. Embedded objects will not be transferred to the database field.

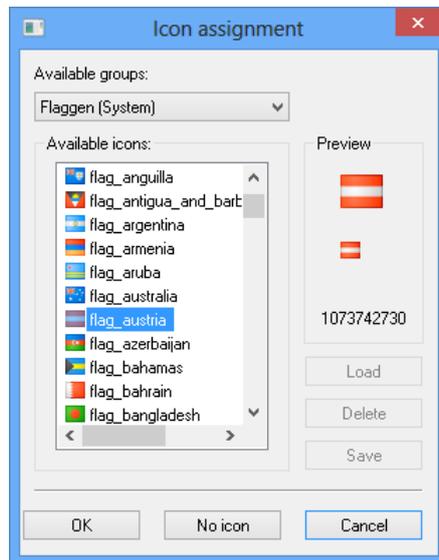
An additional text field is required for the enaio® Outlook add-in that provides functions enabling users to transfer e-mails to the archive. This text field is used to manage the location of transferred emails.

Icons

Folder and register types have standard icons assigned, module types have module-dependent standard icons.

You can assign any other icon than the standard icon to the folder, register, or document types.

If you click on the row Icon in the properties window of an object, you open the dialog **Icon assignment**.



Icons are managed in groups. You can assign icons to the user defined groups but not to the system groups.

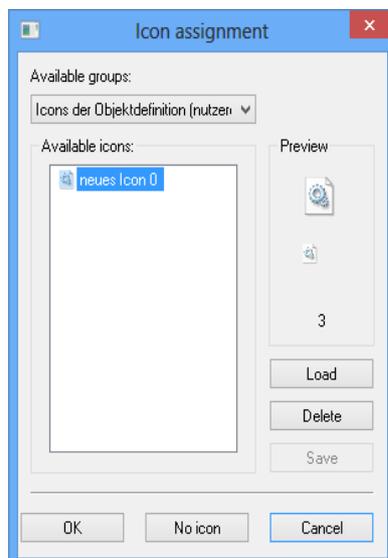
Loading Icons

You can create your own icons or extract them from libraries (*.dll) or programs (*.exe). Bitmaps (*.bmp) can also be loaded. They will be converted. Icons are saved in the database so that they can be accessed from all workspaces.

Follow these steps to load an icon:

1. In the properties window of an object type, click the row Icon.
2. Select a custom group from the list.
3. Click the **Load** button.
4. Via the file selection dialog select an icon, library or program.

The icon or the contained icons will be displayed.



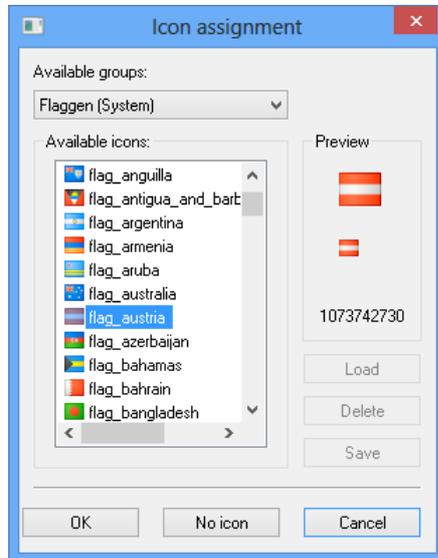
5. Select an icon and click **Apply**.
6. The icon will be added to the group. The names of the icons are automatically managed.
7. Click **Save**.

The icon can be assigned to an archive object and will be saved in the database as soon as you save the object definition.

Assigning Icons

Follow these steps to assign an icon:

1. In the properties window of an object type, click the row Icon.
2. Select a group from the list.
3. Choose an icon.



4. Confirm with **OK**.

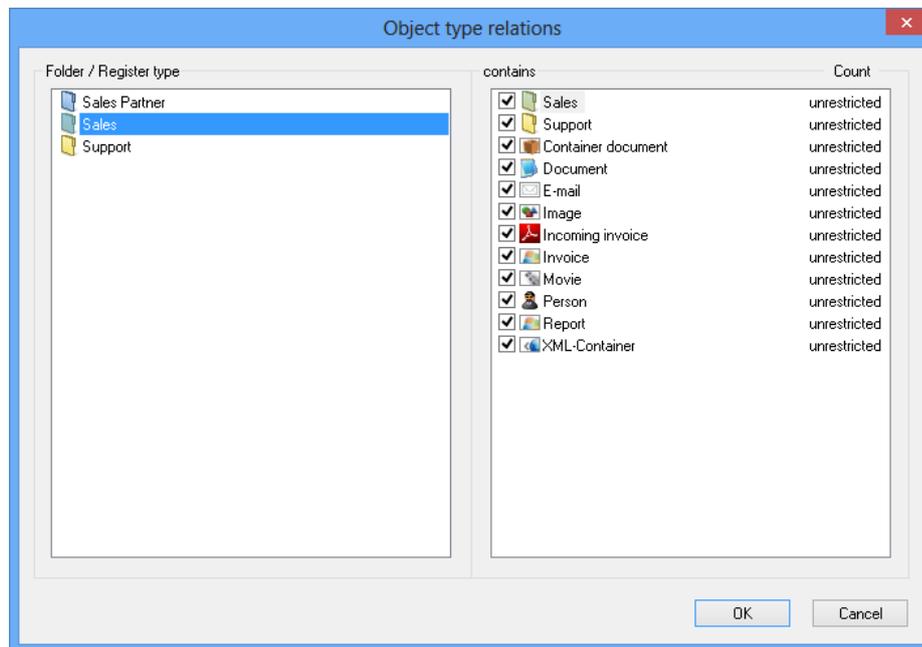
The icon will be assigned to the object type.

In the properties window there is an ID specified for the icon.

Specifying Object Type Relations

Registers and documents are always created by users in folders. Via the object relations you can determine on which level of a folder type a register or document type can be created and how many.

In the workspace, select a cabinet type and via the context menu open the window Object relation.



On the left side, the folder and register types are listed and on the right there are the register and document types listed.

If you mark a folder or register type, you can select the objects from a list of register and document types which are allowed to be created by the user on this folder or register level. Specify the desired number or 'u' for 'unrestricted' using the Number column.

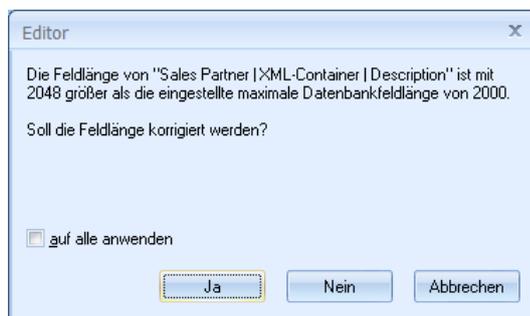
Validating an Object Definition

Before you let users have access to the object definition you should validate it. Especially by copying and importing, data inconsistencies can occur within the object definitions.

The validation of the object definition checks the following points:

- § **Maximum database field length**
Checks if the maximum database field length for text fields which was defaulted has been exceeded by the dialog elements.
- § **Minimum and maximum dialog size**
Checks if the minimum or maximum dialog size for objects has been exceeded.
- § **Catalog syntax**
Checks if the syntax of the catalog entries is correct.
- § **Cross references between catalogs**
Checks if the specified catalog exists in the catalog reference.
- § **The property 'Controlled by cross check'**
Checks if the referred to dialog element in a cross reference between dialog elements with the property 'Controlled by Cross-Check' exists.

If an inconsistency is registered there will a notification will be displayed.



You can choose to let the entry be corrected, to let all similar entries of this type be corrected, to not let the entry be corrected, to not let all similar entries of this type be corrected or to cancel validation.

The validation is started via the entry **Validate object definition** from the menu **File**. The current editing status of the active object definition will be validated.

If you let the object definition be changed by the validation, you must save and then reload the engines so that the changes are available to the user the next time enaio® client is started.

If you let the database field length be changed you must update the corresponding tables afterwards.

Validation also opens the performance wizard (see below).

Validation does not check if the internal name contains special characters which are not valid in the active version.

This active validation is only available to users with appropriate rights. Users with limited rights, e.g. who can only edit catalogs, can perform a passive validation. The results are displayed in the output area. If a log file was specified on the **Output logging** tab, such log file is additionally created.

The Performance Wizard

The performance wizard checks the consistency of the property 'Case-sensitive' and the respective database property.

For databases it can be determined whether capitalization should be noted in the queries or not.

If your database is not case sensitive it can make sense to assign the property 'Case-sensitive - No' to the dialog elements. If dialog elements have this property the query will be restated with the redundant feature 'Upper'. Large delays can be caused due to the fact that databases usually disregard the database index for such queries.

It only makes sense to give the text fields of database type 'all characters' or 'letters' the property 'Case-sensitive - No' to make the queries easier, if your database usually notes capitalization in queries. However, the database index is usually not influenced by this setting.

The performance wizard checks if the database matches case or not. If the database does not note capitalization all dialog elements with the property 'Case-sensitive - No' will be displayed and you can select to change the property for every single dialog element or for all elements at the same time.

Afterwards you must save the object definition so the changes are applied. These changes have no effect on the database tables. They are already created and do not have to be updated.

You can open the performance wizard independent from the validation process via the **File** menu.

Defining Dialog Elements

Dialog Elements - Introduction

You can create dialog elements for the dialogs of folder, register and document types.

There are elements for the graphic properties and elements for the indexing of folders, registers and documents.

The indexing elements have:

- § Graphic properties – the size and position in the dialog,
- § Database properties – a database type and a field length,
- § And properties which support the user when indexing or searching.

The database properties determine which tables are created in the database. Editing of these properties is very restricted.

With the field length you specify a maximum value for the length of the indexing of a field. enaio® editor operates with a defaulted value of 248 characters for text fields which is supported by every database. If the database allows a larger value you can increase the value. The maximum number is found in the documentation of your database. A larger number is entered via the menu **File / settings** on the tab **Maximum database field length**. The maximum permitted value is restricted to 8192 characters.

The dialog has a maximum of 240 dialog elements. If you set up a dialog element of type 'Page control', you can additionally add 240 dialog elements on every page.

The name of a dialog element can have a maximum of 240 characters. The character combination ']]>' is not permitted to be used for names.

Creating Dialog Elements

The dialogs are displayed on the work area. You select the dialog elements via the toolbar **Dialog elements** and position them in the dialog.

The database properties and the properties for indexing are determined via the properties window. There you assign the catalogs and AddOns to the dialog elements. The properties of the dialog elements are not always visible for the user. Therefore use the help of the tooltips to inform yourself about the features of the dialog elements.

If you close a dialog you can select whether you want to adopt the changes or drop them. If you apply the changes they will be temporarily saved. If you want to permanently adopt the changes, save the object definition.

Dialog Element Types

The following element types are available:

- § TEXT BOX
- § Check box
- § Radio button

- § Group field
- § Graphic
- § Static text
- § Button
- § Table
- § Page control
- § Web control

TEXT BOX



The element type 'Text field' is a field where the user enters values for indexing and for queries.

For text fields, you specify the type of database field and the length.

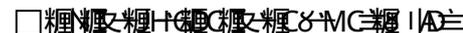
You can specify properties and set up catalogs to support the user in indexing or to make queries easier (see 'Properties of Text Fields').

The text fields can be single or multiline and the name can be visible or invisible in the form.

Check box



Check boxes can be selected



The associated database field is predefined: the type is 'small integer,' the length is '1.'

The hit list in enaio® client has a standard value Yes for selected and No for not selected.

You can edit this default via the properties window (see 'Properties of Checkboxes').

Radio button



Via radio buttons the user can choose exactly one option from the specified ones by selecting it.

The options are set up as separate dialog elements. The associated database field is predefined: the type is 'small integer,' the length is '1.'

In order to assign radio buttons to each other they have to be created within a group field. First you create a group field and then you insert the radio buttons into the group field one after the other (see 'Properties of Radio Buttons').

In the hit list of enaio® client the name of the group field will be used as the column name and the name of the selected radio button will be displayed as the value.

Group field



Group fields are used to optically combine other fields in one frame so that large indexing and search forms remain clear. They have no assigned database field.

Group fields are also used to assign radio buttons to each other (see 'Properties of Group Fields').

Graphic



With the element type 'Image' you can attach graphics to a dialog. Images do not have an assigned database field.

Images have to be available in the Bitmap-Format (*.bmp). The path is specified in the properties window (see 'Properties of Images').

Static text



The element type 'Static text' is also a dialog element which is only used for the design of the dialog. There is no database field assigned. You use it to determine a location for text in the dialog (see 'Properties of Static Text').

Button



Users can execute 'Events' using buttons. 'Events' are individually designed for you by OPTIMAL SYSTEMS or are created using enaio® editor-for-events.

Table



On an indexing form you can manage the information in tables. You determine the number of columns for the table and assign a database type and database length for every column (see 'Properties of Tables').

Page control



The element type 'Page control' is used for designing the dialog. You can use it to combine tabs and other dialog elements in a clear manner.

On a tab of the element type 'Page control' you may create a maximum of 240 dialog elements, however, no further elements of type 'Page control'.

Web control



You can place windows on an indexing form in which URLs are displayed (see 'Properties of Web Controls').

You can enter URL for the following contexts:

§ Create, edit indexing, query, view

Scripts can be used to display document files or previews, too.

Creating Dialog Elements

You select the element type from the toolbar **Dialog elements** with your mouse. Then click any place on the opened dialog.

An opened dialog in the work area will be displayed according to the grid settings (see 'Layout of Forms').

These are the steps to create dialog elements:

1. Open the dialog of an object type from the workspace by double-clicking or via the entry **Edit** from the context menu (menu key).
2. The dialog will be opened in a separate window. In the title row the cabinet and the name are specified.
3. Click the requested dialog type of the toolbar **Dialog elements**.
4. The cursor is used as a positioning tool.
5. With the left mouse button click anywhere in the dialog.

The dialog element will be created with a standard size and name. It will be selected. The properties of the dialog element are shown in the properties window. Open the window by double-clicking on the archive object in the workspace.

If you hold the left mouse button, you can change the size of the dialog element.

If you hold down the mouse button and press **Ctrl**, you can create multiple dialog elements of the same type one after the other.

Via the properties window, with the mouse and by using the tools from the **Layout** toolbar you can position and arrange the dialog elements.

Dialog elements which are opened in the work area can be copied or cut and then inserted on to the same or a different dialog (see 'Copying, Inserting, and Deleting Dialog Elements').

If you create or edit the properties of a dialog element, there will be '*' appended in the title row of the form and the properties window until you have adopted the change.

When you close the dialog or the properties window you will be asked if you want to adopt or drop the changes.

If you adopt the changes they will be temporarily saved. If you want to adopt changes permanently, save the object definition.

New dialog elements are shown in the tree structure in the workspace after you adopt them. Until you save the object definition they will have a '*' appended. If tables need to be adjusted, a '|' is appended.

Deleting Dialog Elements

Dialog elements are deleted via the context menu of the dialog element in the workspace or context menu of the selected dialog element in the work area. If you adopt changes when closing the dialog they are temporarily saved. Save the object definition afterwards.

If you delete dialog elements for which a column has been created in the corresponding database table of the object type, the column will not be deleted and the data will remain in the database. The data can no longer be accessed via enaio® clients, not even via editing histories.

Properties of Dialog Elements

Depending on the type, dialog elements have different properties. The database properties of the text fields, and indexing and query properties are determined via the properties window, the graphic properties are set up easiest using the mouse and the layout features.

Static text, graphs and buttons only have graphic properties.

All dialog elements on a dialog have a tab order assigned. This order can be determined via the layout features.

The properties window is opened by double-clicking the archive object in the workspace. The properties window of the according dialog will be opened.

In the top section of the properties window you find a list of all of the dialog elements of the archive object, from which you can select a dialog element to display the properties for.

If you have selected multiple dialog elements on the workspace (see 'Selecting and Positioning Dialog Elements'), the properties window will only list the properties that all of the dialog elements selected have. If a given property varies among the selected dialog elements, you will find three asterisks in the corresponding row.

Dialog elements can be given internal names. You require internal names when you use scripts, different language versions, and AddOns which refer to other dialog elements. A maximum of 100 characters are allowed. For the first character of an internal name the characters 'a'-'z', 'A'-'Z' and the character '_' are allowed. For the other characters numbers and the dot are also allowed.

Special characters can be used for naming the dialog elements. However, if you use scripts to refer to dialog elements containing special characters, errors may occur. Try to avoid special characters or use internal names for referring to dialog elements.

Dialog elements which are managed by automatic features can be created with negative coordinates so that they are created in an invisible section.

Within the name for dialog elements the character '&' on dialogs has a special feature. If you place a '&' in front of a letter the user can use the keyboard shortcut **Alt-letter** to get to the dialog element. For dialog element 'Button', this will automatically execute the button's event.

If you wish to use a '&' within the name of a dialog element, please enter '&&'.

enaio® client uses **Alt-A** on search forms for the feature **Start query** and **Alt-E** on the data dialogs for the feature **Save**. If you assign your own shortcuts for field names, these have priority.

When used outside of dialogs, the '&' character is part of the dialog name and will be displayed as a character in other situations, for example in a hit list or as a title for a column.

Properties of Text Fields

For text fields can you specify the database properties, for all other element types the properties are defaulted. Catalogs and AddOns can be assigned to text fields. The name and the element can be positioned in the dialog independent on each other.

Text fields have the following properties:

Name

The name of the element will be used as a label of the dialog. A maximum of 240 characters, including special characters, is allowed.

Database Field Type / Field Length

Via the type of a database field and the length, you can specify which and how many characters the user can enter in the text field.

As for database types, possible entries can be made with regular expressions (see below).

For the maximum valid amount of characters of a text field enaio® editor uses a default value of 248 characters. If your database allows longer fields you can use the **File / settings** menu to enter a larger number of allowed characters in the dialog **Maximum database field length**. The enaio® editor automatically corrects the maximum database length for field lengths which exceed the maximum defaulted value.

The following database types can be selected:

§ All characters

All ASCII-characters are allowed.

The maximum length is dependent on the database.

The internal database type is CHAR.

§ Letters

All letters A-Z, a-z, spaces, and special characters are allowed.

The maximum length is dependent on the database.

The internal database type is CHAR.

§ Uppercase letters

All letters A-Z, spaces, and special characters are allowed.

The maximum length is dependent on the database.

The internal database type is CHAR.

§ Date

The maximum length is 10 characters.

The internal database type is DATE.

The date is entered in a date field in one of the following formats:

01.02.1999, 1.2.1999, 1.2.99, 02021999, 020299.

For two digit values larger or equal to 50 a 1900 is added. If the value is less than 50, 2000 will be added.

Date fields can be defaulted with the current date.

For date fields an interval query is possible.

§ Date/Time (Time stamp)

The time stamp consists of a date and a time and is internally saved in seconds after 1.1.1970 00:00:00.

The internal database type is INTEGER.

Date/Time fields can be pre-defined via the feature **Current date/current time**.

An interval query is possible for Date/Time fields.

§ Time

The time is in hours, minutes and seconds.

The internal database type is LONGINTEGER.

The separator between hours, minutes, and seconds is a semicolon.

Time fields can be pre-defined with the feature **Current time**.

For time fields an interval query is possible.

§ Decimals

The preset length is 15 characters: 13 digits before the decimal point and two decimal places after. Plus a dot or comma as a separator. Decimal numbers always have a maximum of two decimal places.

The internal database type is DECIMAL.

The maximum length is dependent on the database.

§ Integers (numeric)

Integers composed of the characters 0-9 are allowed.

The defaulted length is 9 characters.

The internal database type is INTEGER. The maximum size is dependent on the database. For INT4 it is 2,147,483,647.

Please note that the zero is after the nine when sorting by numbers.

Select alphanumeric numbers if the sorting is important for the dialog element.

A symbol for the number grouping is assigned in the default by entering a constant.

For number fields an interval query is possible.

The maximum length is dependent on the database.

§ Integers (alphanumeric)

Integers composed of the characters 0-9 are allowed.

The maximum length is dependent on the database.

The internal database type is CHAR.

Alphanumeric integers should be selected if sorting is important for the dialog element. When sorting alphanumeric integers the zero comes before the one.

A symbol for number grouping and interval queries are not possible.

§ only [a/s/f] for patient type

Only 'A,' 'S' or 'F' are allowed.

(ambulatory, stationary, external patient)

§ only [y/n] for questions

Only 'Y' or 'N' are allowed.

(yes, no)

§ only [l/r] for side

Only 'L' or 'R' are allowed.

(left, right)

§ only [m/f] for sex

Only 'M' or 'F' are allowed.

(male, female)

If you change the database type for a field a table column is created for it when the database is updated. If data has already been stored, it will be lost. The field length can be enlarged but not reduced.

Multi-Field

Multi-fields are standard fields for P-, X- or D-documents which are created in the section for multi-fields on the lower frame of a dialog. Via multi-fields every page of a document can be indexed individually (see 'Multi-Fields').

Label x-Position / Label y-Position

Name label and the entry field of a text field can be positioned in the dialog independent of each other.

The 'Zero point' of the dialog is the upper left hand corner; the reference point of the label field is the upper left hand corner of the label field. Values between '-2000' and '2000' are allowed.

For labeling the system font is always used.

Label Width / Label Height

The width and the height of the label field are the reference points of the layout feature. Parts of the label which are located outside of the label field will not be visible. Within the label field, the label is always aligned on top. Label fields are not transparent. Values between '0' and '2000' are allowed.

Label Alignment

The label within the label field can be set on the right, centered or on the left.

Element x-Position / Element y-Position

You enter the coordinates for the top left hand corner of the entry field in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

For entries in the entries fields, the system font is used. The width and the height you enter are proportional to the system font type. For the height of an index data field we recommend a value of 12 units. The width is best adjusted using the sample index data in the test mode. Values between '0' and '2000' are allowed.

Restriction due to a form

A form restriction structures the indexing. For example, if a file number always has the same structure you can determine where the user enters the characters in the dialog.

Restriction for a dialog only applies to indexing dialogs not to search forms.

How to Define a Form

If you want to define a form, select the entry by form from the list in row Restriction. Enter the form definition in the row Form.

For every space in the index data field you set up a placeholder or fixed character in the row Form.

These placeholders can be used:

Placeholder	Allows the characters
9	Integers (0-9)
N	Integers (0-9) and letters (a-Z)
X	All ASCII-characters
A	Letters (a-Z)
U	Uppercase letters
L	Lowercase letters

All other characters are fixed and defaulted in the dialog. If you want to use one of the characters which is used as a placeholder as a fixed character in the dialog, you must add a forward slash '/' in front of it.

Example of a self-defined form



The slash in front of the letter 'A' which is at the second position and could be a placeholder determines that it is a fixed character. The third and fourth position are also fixed characters. The sixth position has the placeholder 'U' for a capital letter. The slash at position seven determines that the slash at position eight is a fixed character. At position nine and ten are placeholders for the integers. At position eleven there is a fixed character. At position twelve and thirteen there are again placeholders for integers.

On the indexing form in the client the form looks as follows:

FileNumber AZ: / / . _

After you have applied the changes you can check the dialog in test mode.

Regular expression

You can use regular expressions to limit the entry options for users. If you enter a regular expression, the user will receive an error message in the client if the text field entry does not match the regular expression.

enaio® editor does not check entered regular expressions for consistency. Regular expressions are saved in a database column, the maximum length is equal to the maximum length of a database field.

Required field

A required field must be filled out by the user when indexing. It is underlined in the indexing and search form.

Supervisor field

A Supervisor field can be edited while indexing by a user with the required system role.

Read-only

Read-only properties cannot be changed by events.

Fields with the property read-only cannot be edited by any user. They can however be indexed by AddOns for example.

Read-only after archiving

Fields with the property read-only after archiving cannot be edited after they have been archived in a permanent manner.

Read-only after initializing

An entry in fields with the property read-only after initializing cannot be edited after it has been saved.

Automatic Copying

With fields which you have assigned the property automatic copying, the users can create copies of documents. The document cannot be located in the filing tray. The copies are created as soon as the user enters an integer in this field using the feature **Edit data** or enters integers which are separated by a semicolon.

Example: 30-90

enaio® creates copies of this document which have the field property automatic copying and are indexed with consecutive integers from 30 to 90.

Fields with the property automatic copying are not especially identified.

With Leading Zeros

Fields with the property with leading zeros are filled up with leading zeros until the database field length is completed if the user has entered fewer digits than are specified for the database field length.

Key field at transfer

enaio® checks for key fields if the entered indexing is already used in the database for this field. If this is the case, the user receives an error message in the status bar.

If one or multiple fields have this property, only the value in one of the key fields has to be new.

Key fields are cursive in the indexing and search forms.

Controlled by Cross Check

Fields with the property Controlled by cross check are filled out via structure tree catalogs or add-ons (see 'AddOns for Text Fields').

Linked to a W-Document Template

You assign applications and templates to the W-Documents via enaio® administrator. You can define multiple templates for an application and let the user select which one he

wants to use. If you assign the property Linked to W-Document template to a text field of a W-Document, the selection dialog for templates does not appear when you create a W-Document but the template which is identical to the entry in the field is used.

Case-sensitive

The property Case-sensitive applies to queries for text fields of database types all characters or letters. If Case-sensitive – yes is specified for a query, the search will be case sensitive for this field.

Only for text fields of the database type all characters or letters the query can be simplified if the user does not require case sensitivity for the query expression. The property Case-sensitive - no can cause remarkably longer query times.

For databases, you can configure whether queries should be case-sensitive. If the database does not pay attention to case, then the property Case-sensitive – No is redundant. Please use the performance wizard (see 'The Performance Wizard') to check the database setup and to check the field properties for case sensitivity.

Defaults and Constants

Fields can be preset with a constant or with a function.

Functions can be overwritten by the user. In the row Functions, select one or more of the following features:

- § Current date
Inserts the current date.
- § Current date and time span
Defaults to the date consisting of the configurable time span added to the current date in days
The number of days is entered in the following Constant row. A negative value is invalid.
- § Current object-ID
Defaults to the object-ID of an archive object.
- § Current user
Defaults to the user name of the OS user currently in the text field.
- § Current user full name
Defaults to the full name of the OS user currently in the text field.
- § Current time
Defaults to the current time.
- § Current date / current time
Defaults to the current date / current time for the date/ time field.
- § User(u)
User name as entered by the right group AddOn. A '(U)' is appended to a user, to distinguish between user names and group names.

The constant can also be overwritten by the user. In the Constant row, enter the desired value. A maximum of 76 characters, including special characters, is allowed.

For fields of the database type 'Integer,' a symbol can be specified as a 'Constant' for number grouping with following syntax:

Group width|Symbol

Possible values for group width are: 0,1,2,3,4,5,6,7,8,9,32

As a value for the symbol, all characters and also multiples can be specified.

Examples:

3|, 1,000,000

6|- 1-000000-000000

Integers with up to 255 digits can be formatted like this.

Only integers in fields of database type Integer can receive a symbol for number grouping.

Catalogs and AddOns

Catalogs and AddOns can be assigned to text fields. Select a catalog type or AddOns from the list offered in this line.

The properties window will have an additional tab assigned. On the tab you can edit the catalog entries (see 'Catalogs for text fields') and configure add-ons (see 'AddOns for Text Fields').

Automatically execute add-on

AddOns can be automatically executed upon saving. This feature only makes sense for AddOns without user action, for the ID AddOn for example.

Full text export

For object types with the property 'Full text indexing – automatic' or 'Full text indexing – not automatic' you can individually select whether the indexing should be automated or not for every single dialog element.

Search group

A search group has an additional query feature. If you insert a query expression in a text field that is part of a search group, the search will be carried out over all fields of the group.

Assign text fields to a search group by specifying the same search group name for the text fields of an object type.

A maximum of 30 characters is allowed for the name of a search group. For the first character the letters 'a'-'z', 'A'-'Z' and the underline '_' are allowed, other characters, digits and full stops are allowed for the following characters.

Search group fields are not indicated on forms. Inform your user about this property.

Auto-complete

Auto-complete can be switched on for text fields from the 'all characters,' 'letters,' and 'upper case letters' database types. The user is then given a list of suggestions based on existing entries when they enter three characters or more when searching. The rights

system is taken into account, however the other fields already filled out are not considered when determining the suggestions.

This function requires a database index for the field.

Auto-complete is not available for multi-line text fields or text fields with AddOns.

This function can be switched off for all database catalogs by adding an entry to the `as.cfg` file, located in the `\etc` directory of the data directory:

```
[SYSTEM]
```

```
USETYPEAHEADONIDXCHARFIELD=0
```

Text field type

Text fields can have one or multiple lines. A multiple line text field makes sense for longer texts, for example for freely written text passages.

Dialog element visible

You can hide the dialog element. This property can be edited context independently via events.

Label visible

You can hide the label in the dialog. However, it will be used as the name of a column in the hit list in enaio® client.

Via the menu **File/Settings/Miscellaneous** you can set up whether the names in enaio® editor for the layout should be visible. In test mode they will not show.

Tab position

The tab position of the element on the form is entered into this line. The order is edited via the layout features (see 'Tab Position of Dialog Elements').

Tooltip Text

The entered text will be displayed as a tooltip as soon as the user hovers the mouse above this element. A maximum of 123 characters, including special characters, except for the pipe sign, are allowed.

Text color

The text color for the indexing entries is selected via the standard text color selection dialog.

Catalog tooltips

You can assign a tooltip to the catalog and add-on button. A maximum of 123 characters, including special characters, except for the pipe sign, are allowed.

Properties of Checkboxes

In contrast to the text fields, you cannot hide the name of check boxes or position them in the dialog independent of the check box. They are always placed on the right of the check box.

The database properties are automatically assigned, in the multiple section, check boxes are not allowed.

Name

The name of the element will be used as a label of the dialog. A maximum of 240 characters, including special characters, is allowed.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the element in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

Width and height of the element fields are the reference points for the layout features. The selectable field and the name will always be placed left and centered horizontally. Values between '0' and '2000' are allowed.

Supervisor field

A check box with the property Supervisor field can only be selected by users which have the corresponding system role.

Read-only

Read-only properties cannot be changed by events.

Checkboxes with the property read-only cannot be selected by any user.

Read-only after archiving

Checkboxes with the property read-only after archiving cannot be edited after they have been archived in an audit-proof manner.

Read-only after initializing

A checkbox with the property read-only after initializing cannot be changed anymore after it has been checked.

Key field at transfer

enaio® checks for key fields if the entered indexing is already used in the database for this field. If this is the case, the user receives an error message in the status bar.

If one or multiple fields have this property, only the value in one of the key fields has to be new. Check box fields should only have this property in combination with other fields.

Key fields are cursive in the indexing and search forms.

Controlled by Cross Check

Checkboxes with the property Controlled by cross check are filled out via add-ons (see 'Quickfinder AddOn).

Defaults

Initially, a check box is always either not set or set. You determine the default for the checkbox.

Display value when checked

Displays the value in the hit list when the check box is set. The default is 'Yes.'

Display value when not checked

Displays the value in the hit list when the check box is not set. The default is 'No.'

Dialog element visible

You can hide the dialog element. This property can be edited context independently via events.

Tab position

The tab position of the element on the form is entered into this line. The position can be edited using the layout functions.

Tooltip Text

The entered text will be displayed as a tooltip as soon as the user hovers the mouse above this element. A maximum of 123 characters, including special characters, except for the pipe sign, are allowed.

Properties of Radio Buttons

Radio buttons must always be positioned within group fields and require a tab position which is lead by the first group field.

Similar to check boxes, radio button labels cannot be hidden or positioned independently from the button. They are always placed on the right of the check box.

The database properties will be set automatically, in multiple field sections, radio buttons are not possible.

Name

The name of the element will be used as a label of the dialog. A maximum of 240 characters, including special characters, is allowed.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the element in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

Width and height of the element fields are the reference points for the layout features. The selectable field and the name will always be placed left and centered horizontally. Values between '0' and '2000' are allowed.

Required field

If a radio button receives the property mandatory field, the user must select an option.

Required fields are underlined in the indexing and search forms.

Supervisor field

If a radio button receives the property Supervisor field, only the user that has all roles can select an option.

Read-only

Read-only properties cannot be changed by events.

If a radio button receives the property read-only, no user can select an option.

Read-only after archiving

After the permanent archiving this property cannot be edited anymore.

Read-only after initializing

If a radio button receives the property read-only after initializing, no more changes can be made after this option was selected.

Key field at transfer

enaio® checks for key fields if the entered indexing is already used in the database for this field. If this is the case, the user receives an error message in the status bar.

If one or multiple fields have this property, only the value in one of the key fields has to be new. Radio buttons should only receive this property in combination with other fields.

Key fields are cursive in the indexing and search forms.

Constant Value

If a user creates a new archive object, one of the radio button is always created with the value '1' assigned to as a constant by default. If all radio buttons have the value '0' assigned, the one with the lowest tab position is activated by default.

Dialog element visible

You can hide radio buttons. This property can be edited context independently via events.

Tab position

The tab position of the element on the form is entered into this line. The position can be edited using the layout functions. Radio buttons which are assigned to each other need a continuous sequence of tab positions and must immediately follow the group field.

If the table is updated, the relative order of the tab positions between a group field or radio buttons can only be edited if no data has been entered. Otherwise you risk data loss or data may be used incorrectly.

Tooltip Text

The entered text will be displayed as a tooltip as soon as the user hovers the mouse above this element. A maximum of 123 characters, including special characters, except for the pipe sign, are allowed.

Properties of Group Fields

With group fields you can clearly structure forms. Related radio buttons have to be placed within a group field.

Name

The name of the element will be used as a label of the dialog. A maximum of 240 characters, including special characters, is allowed.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the element in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

Width and height of the group field are adjusted to the element with the layout features of the group field. Values between '0' and '2000' are allowed.

Dialog element visible

You can hide group fields.

Label visible

You can hide the label of a group field.

Via the menu **File/Settings/Miscellaneous** you can set up whether the names in enaio® editor for the layout should be visible. In test mode they will not show.

Tab position

The tab position of the element on the form is entered into this line. It does not matter to the user of enaio® client, but is important for the assignment of radio buttons.

The position can be edited using the layout functions.

Properties of Tables

Via the tab **Properties**, you can specify the graphic features of the table, on the tab **Table** you create tables. On the dialog a placeholder will be assigned to the table. The table will only be displayed in test mode.

Properties tab:

Name

The name of the table will always show on the left upper side of the table. A maximum of 240 characters, including special characters, is allowed.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the element in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

You specify the width and the height of the table in the dialog. Values between '0' and '2000' are allowed.

Read-only

Read-only properties cannot be changed by events.

Fields with the property read-only cannot be edited by any user. They can however be indexed by AddOns for example.

Multi-selection

More than one table row can be selected.

Dialog element visible

You can hide tables. This property can be edited context independently via events.

Full text export

For object types with the property 'Full text indexing – automatic' or 'Full text indexing – not automatic' you can individually select whether the indexing should be automated or not for every single dialog element.

Tab position

The tab position of the table on the form is entered in this row. The position can be edited via the layout features.

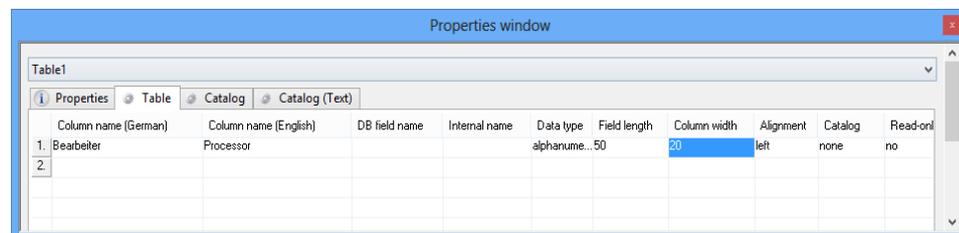
Tooltip Text

The entered text will be displayed as a tooltip as soon as the user hovers the mouse above this table. A maximum of 123 characters, including special characters, except for the pipe sign, are allowed.

Tab Table

Here you can specify a column name for each column of the table, optionally an internal name from a list box as data type 'alphanumeric', 'numeric', 'date' or 'decimal', enter the maximum number of characters which the user can enter in the cell in Field length, specify the layout width in Column width and select an alignment for the contents of the column from the list field. Columns can be set to read-only. The DB field name cannot be edited.

If you select a column number you can use the up and down arrow keys to change the order. Columns are deleted by selecting and deleting the column.

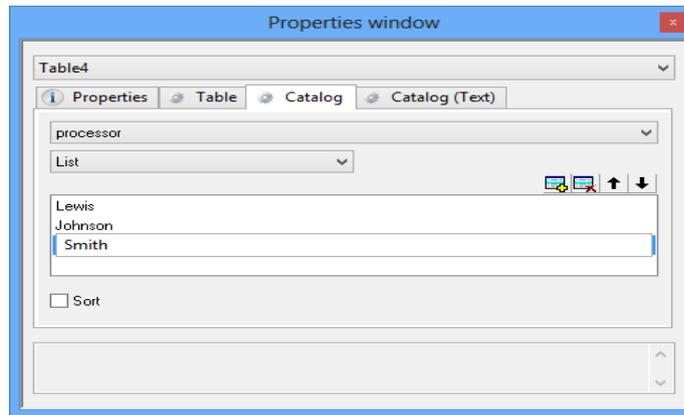


	Column name (German)	Column name (English)	DB field name	Internal name	Data type	Field length	Column width	Alignment	Catalog	Read-only
1.	Bearbeiter	Processor			alphanume...	50	20	left	none	no
2.										

Use the Catalog column to specify whether you want to set up a list, database, tree, or hierarchy catalog for table columns.

Assigned database catalogs will display all values without any limitation.

You can specify the catalog entries via the corresponding tabs (see 'Catalogs for text fields').



Queries of Tables

For a query of table fields, only the column in which the user enters the search expression is searched. Only one search expression can be entered in the first column.

Only tables with one column can be configured so that several search expressions are permitted in several cells in the column. The search expressions are linked with AND for the query.

For this, you need to add an entry with the following structure to the `as.cfg` file, located in the `\etc` directory of the data directory:

```
[SYSTEM]
MULTIREQUESTTABLECONTROLS=TablenameDatabaseName
```

The table name is the table name of the object type; the database name is the database name of the table field.

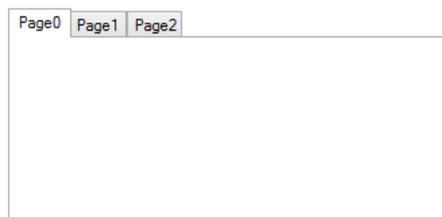
If you want to specify more than one table, separate the names with a semicolon.

Example:

```
[SYSTEM]
MULTIREQUESTTABLECONTROLS=object9list1;stamm5list2
```

Properties of Page Controls

With page controls you can clearly structure forms. Page controls can contain multiple pages on which you can align dialog elements sorted by topic.



No page controls can be created on a tab.

Placeholders will show for the page controls in test mode.

On tabs, dialog elements with the property 'Controlled by cross check' can only refer to elements which are also assigned to the same page control and the same page. Dialog elements which are not assigned to a specific page control cannot be referred to by dialog elements that are assigned page controls.

Pages

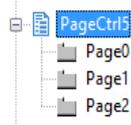
If you create a dialog element of type 'Page control', the first page receives the name 'Page 1'.

The name of the page control, the internal name and the graphic properties are entered via the properties dialog. Every page can have an icon assigned to the tab.

If you select the name of the page in the dialog you can add and delete pages via the context menu. The name of the page is entered via the properties dialog. The width and the height are automatically adjusted by the page control. If you use the properties window or the context menu to change the page numbers the other page numbers are automatically adapted.

On the individual pages you can then create the dialog elements.

In the workspace the page control is shown. In the search tree to which the pages are assigned, the dialog elements are assigned to the pages.



On the page the tab position of the elements is relative to the page and begins with '0'.

Properties of Web Controls

You can integrate URLs in indexing forms with a web control.

Enter the size and position for the display window and URL for the corresponding context.

Name

The specified name will not be displayed on the form.

Element x-Position / Element y-Position

You can specify the coordinates for the upper left hand corner of the window on the form. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

You specify the width and the height of the window on the form. Values between '0' and '2000' are allowed.

Standard-URL

The URL specified here is used in all contexts except in the following. The entry can be left blank if you specify URL for all contexts.

Creation-URL

This URL is displayed when the indexing form of an object is displayed during creation.

Editing-URL

This URL is displayed when the indexing form of an object is displayed during editing.

Query-URL

This URL is displayed when the search form is opened.

View-URL

This URL is displayed when the indexing form of an object is displayed as read-only.

Dialog element visible

You can hide web controls.

Tab position

The tab position of the window on the form is entered in this row. The position can be edited using the layout functions.

Properties of Images

Images are always displayed in original size in the dialog. They receive an internal tab position which does not matter to the usage of enaio® client by the user. An image can be specified for each language.

Name

The specified name will not be displayed on the form.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the graph in the dialog. Values between '-2000' and '2000' are allowed.

Dialog element visible

You can hide graphics.

Image ID

Images must be in bitmap format (*.bmp) with a maximum of 256 colors. Specify a file in the file selection dialog. An image can be specified for each language.

Images are also saved inside object definition files.

Properties of Static Text

Static text can make orientation in the dialog easier for the user.

Name

This text will be displayed in the dialog. A maximum of 240 characters is possible.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the element in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

The width and the height of the element are the reference points for the layout features. The text will be placed on the upper frame of this field. Values between '0' and '2000' are allowed.

Label Alignment

The text can be placed left, right or centered.

Dialog element visible

You can hide static text.

Tab position

The tab position of the static text is irrelevant to the user.

Properties of Buttons

Users can execute 'Events' using buttons. 'Events' are individually designed for you by OPTIMAL SYSTEMS or can be created using enaio® editor-for-events.

If a user opens a data sheet as read-only, events linked to buttons will not be executed.

Name

The name of the button will always be displayed horizontally and vertically centered on the button.

If you enter the character '&' in front of a letter, the user can use the keyboard shortcut **Alt-Letter** to execute the event.

enaio® client uses **Alt-A** on search forms for the feature **Start query** and **Alt-E** on the data dialogs for the feature **Save**. If you create your own shortcuts yours have precedence.

Element x-Position / Element y-Position

You specify the coordinates for the upper left hand corner of the button in the dialog. Values between '-2000' and '2000' are allowed.

Element Width / Element Height

You can specify the width and the height of the button in the dialog. Values between '0' and '2000' are allowed.

Dialog element visible

You can hide buttons. This property can be edited context independently via events.

Tab position

The tab position of the button in the form is entered in this row. The position can be edited using the layout functions.

Tooltip Text

The entered text will be displayed as a tooltip as soon as the user hovers the mouse above this element. A maximum of 123 characters, including special characters, except for the pipe sign, are allowed.

Catalogs for text fields

Text fields can have catalogs assigned. With catalogs, the user selects entries for the indexing from lists.

Assigned catalogs are selected with a button behind the field on the indexing and search forms.

The following catalogs can be created:

- § List catalog,
- § Tree catalog,
- § Hierarchy catalog,
- § Database catalog,
- § Structure tree.

You can assign catalogs to text fields which have already been assigned to other text fields. If you edit the catalogs the changes will immediately be applied at all locations.

The catalogs lists for the current object definition are managed on enaio® server. The structure tree catalogs are stored in files named by you and you also specify the path. They have to be located in a directory which can be accessed by the path by all users.

Structure tree catalogs are not shown in test mode.

The entry field and the catalog button can have a tooltip assigned.

Validation checks the catalog entries for correct syntax and correct reference structure. Validate the object definition (see 'Validating an Object Definition'), before you make it available for other users. The character combination ']]>' is not permitted to be used for list entries.

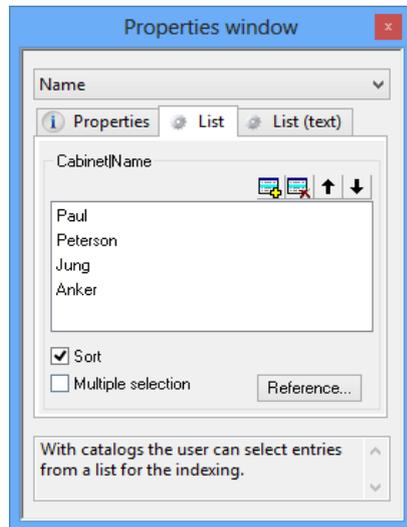
Besides these catalogs, the Catalog AddOn can also be used. It accommodates functions of list, tree and hierarchy catalogs and allows multiple selection for tree and hierarchy functions, too. If you want to access catalog data via scripts you can also use this AddOn.

List catalog

You can create list catalogs from which users can select one value, list catalogs from which users can select several values, and list catalogs from which users can select abbreviations or numbering for the values to apply to the catalog field.

List values can have icons assigned. In enaio® client, hit lists and folder windows of the objects will not be shown with the standard icon but with the icon which is assigned to the list value used.

If you select the **List** item in the Catalog line of the properties window for a text field, the **Lists** tab will be created.



In the modifiable field, enter the list value for each row individually. A new row is added using the  row button or the **Insert** key. List values are moved up or down using the arrow keys or using the keyboard shortcut **Alt-Arrow**. Using the **Del** key or the  delete button, you can delete rows.

The list values will be shown in the currently-displayed order or, if you have checked the Sort check box in the client, they are displayed in alphabetical order.

Abbreviations and numbering to be imported into the catalog fields (instead of the values) are defined with this structure:

```
<Abbreviation>|<List entry>
```

As a separator use the 'pipe' character (|). The values which follow the 'pipe' character will be displayed in the list but not transferred to the catalog field. You can also comment individual list values.

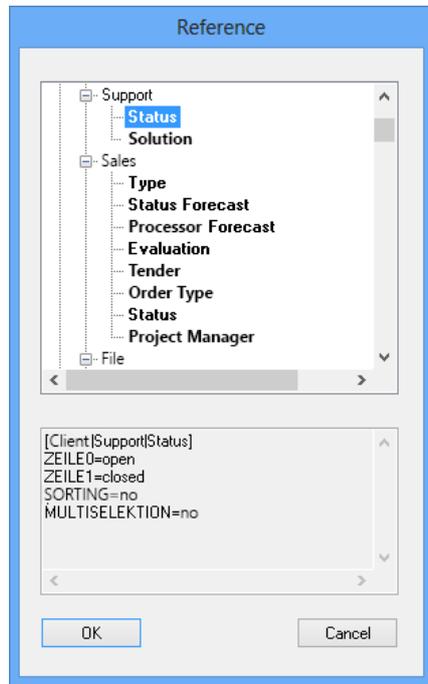
The character 'pipe' (|) must be used for the first list entry. For the following ones it is optional.

List entries which have the character '*' at the end of the row are only shown for the query and not for newly created entries. This lets you mark entries which are no longer allowed for newly created entries, but are still required for search and editing. For editing, the entry is not shown in the list but can be entered manually.

If you mark the check box Multiple selection, the user can select multiple values from the list for the catalog field. Imported values in the catalog fields are separated by semicolons. However, in the query via list catalog with multiple selection, every entry has the placeholder '*' before and after it. This causes the query to become fuzzy.

If multiple selection is possible, the single list entries cannot contain semicolons.

You can assign a list catalog to the catalog field via the button **Reference** which you have already created for a different text field.



In the reference dialog all text fields with catalog lists from the same object definition are displayed. If you mark a catalog field, the list will show in the window. If you press the **OK** button you create a reference to the list of the selected catalog field for the active catalog field.

The list can be edited. Changes apply to all catalog fields which refer to the list. The changes are applied immediately. For references to icon catalogs, you can only make changes in the original icon catalog.

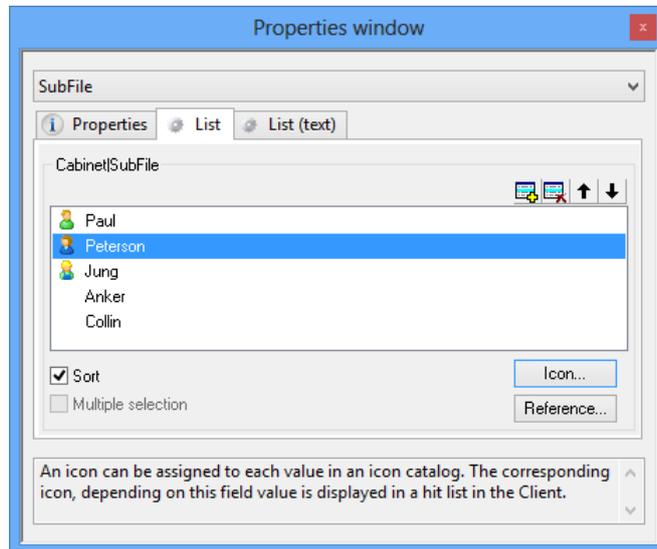
If the dialog for the catalog field is opened and you update for the changes when you close it, the old version will also be stored in the database.

List icons

If you select the catalog type 'List', 'Tree', or 'Hierarchy' for a text field, the property 'Icon catalog' will be inserted into the properties dialog.

On a dialog, only one text field can have the property 'Icon catalog' assigned. A multiple selection for icon catalogs is not possible.

On the **List** tab you find the button **Icon...**



Use the **Icon...** button to open the dialog **Icon assignment**. There you select the intended icon for the list value (see 'Icons').

If you refer to the icon catalog, the entries can only be edited in the original icon catalog.

The archiving status is indicated by an icon in the W-Document variant administration. List icons can be shown by adding the following entry to the file `as.cfg`:

```
[ SYSTEM ]
USESYSICONINVARDLG=0
```

The value '1' switches back to archiving icons.

The file is found in the `\etc` directory of the data directory. You can edit the file with any editor.

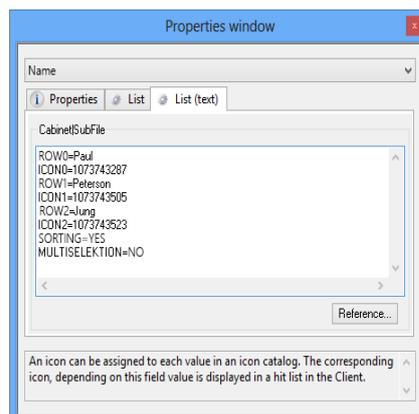
List catalogs for data transfer

With enaio® data-transfer, you can transfer index data for archive objects to a Microsoft Word document.

For index data from list catalogs which contain a list entry and an abbreviation separated by a pipe character, you can use the key word '#LIST#' to apply the list entry instead of the abbreviation for replacement fields.

You can also extend the list catalog with entries which are only used for data transfer:

1. Open the tab **List (text)** in the properties window.



The list is displayed in text format. The entries are numbered.

2. Attach a semicolon and then the value for the data transfer to every row.

Example: Zeile1=B;DB

The fields are numbered consecutively. Assign an entry for the data transfer to the list entry and separate it with a semicolon.

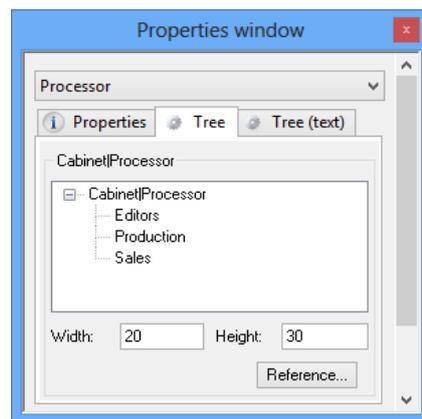
Details for data transfer are found in the manual 'enaio® data-transfer'.

Multiple selection for list catalogs is not be available, even if the function is turned on since semicolons are not allowed for multiple selection of list entries.

Tree catalog

Tree catalogs are list catalogs in which values are created on different levels. This structure facilitates the selection of values from long lists. In contrast to hierarchy catalogs the level names are not adapted to the catalog fields. The user can only select values at the last level.

If you select the **Tree** item in the Catalog line of the properties window for a text field, the **Tree** tab will be created.



On the tab of the selected field name, select New node from the context menu or press the **Insert** key. In the editable field, enter the name of the node on the first level. When you select a node, you can create a node on the next level with the context menu or the **Insert** key.

You can select an already created node and then use the mouse to drag and drop it to a different location. It will be placed beneath the node that you drop it on. All other child nodes will also be moved.

The order of the nodes on the same level can be changed by holding the Ctrl-key and then dragging the nodes. It will be placed below the target node.

Level names are not imported into the catalog field, so identical values from different links cannot be differentiated.

Entries which are followed by the 'pipe' character will be displayed in the catalog like in list catalogs, but are not added to the catalog field. As in list catalogs, entries containing the character '*' at the end of the row are only shown for the query, and not for newly created entries. For editing, the entry is not shown in the list but can be entered manually.

Enter values for the height and width of the dialog in enaio® client.

You can assign a tree catalog to the catalog field via the button **Reference** which you have already created for a different text field. The trees can be edited. Changes will apply to all catalog fields which refer to the tree catalog. The changes are applied immediately.

Validation checks the catalog entries for correct syntax and correct reference structure. Validate the object definition (see 'Validating an Object Definition'), before you make it available for other users.

If you select the option 'Icon catalog' in the property dialog, an icon can be assigned to each entry.

Matchmode

If users enter a value in a field with a tree catalog, and then open the catalog using the catalog button, this value is then selected in the catalog if the value exactly matches a catalog entry. If there are several, then the first catalog entry is selected. If there is no exact match, nothing is selected.

This search behavior can be modified in the catalog. To do so, on the **Tree (Text)** tab enter 'MATCHMODE' with a parameter:

MATCHMODE=0 Default behavior

MATCHMODE=1 If there is no exact match, then the first catalog entry that starts with the value is selected.

You can switch to the next value using the TAB key, and to the previous value using Shift+TAB.

MATCHMODE=2 If there is no exact match, then the first catalog value that contains the value is selected.

You can switch to the next value using the TAB key, and to the previous value using Shift+TAB.

Hierarchy catalog

The hierarchy catalog corresponds to the tree catalog but only the levels descriptions are carried over to the indexing form. That allows you to differentiate identical values from different links.

The values from the hierarchy catalog can be edited by the user in the query in the catalog field, for example a placeholder can be inserted.

In the catalog field the transferred level values are separated, the default separator character is the 'pipe' (|). You can also specify any other character as a separator.

If you use any character other than 'pipe' as a separator, the entries which are followed by a 'pipe', for example in list catalogs, will be displayed in the catalog but not transferred to the catalog field.

Entries of all levels are transferred from the hierarchy catalog. If you select the option **Allow intermediate levels** the user can only transfer entries from the first level to the selected intermediate level.

As in list catalogs, entries containing the character '*' at the end of the row are only shown for the query, and not for newly created entries. For editing, the entry is not shown in the list but can be entered manually.

A hierarchy catalog is created in the same way as a tree catalog.

You can assign a hierarchy catalog to the catalog field via the button **Reference** which you have already created for a different text field. The hierarchy catalog can be edited. Changes

will apply to all catalog fields which refer to the hierarchy catalog. The changes are applied immediately.

Validation checks the catalog entries for correct syntax and correct reference structure. Validate the object definition (see 'Validating an Object Definition'), before you make it available for other users.

If you select the option 'Icon catalog' in the property dialog, an icon can be assigned to each entry.

Matchmode

If users enter a value in a field with a hierarchy catalog, and then open the catalog using the catalog button, this value is then selected in the catalog if the value exactly matches a catalog entry. If there are several, then the first catalog entry is selected. If there is no exact match, nothing is selected.

This search behavior can be changed in the catalog in the same way as for the tree catalog. To do so, on the **Hierarchy (Text)** tab enter 'MATCHMODE' with a parameter:

MATCHMODE=0 Default behavior

MATCHMODE=1 If there is no exact match, then the first catalog entry that starts with the value is selected.

You can switch to the next value using the TAB key, and to the previous value using Shift+TAB.

MATCHMODE=2 If there is no exact match, then the first catalog value that contains the value is selected.

You can switch to the next value using the TAB key, and to the previous value using Shift+TAB.

Database catalog

With the button **Database catalog**, the user can start a query. A list containing all values which have been entered into this field for indexing is created and opened. The user can restrict the query by entering values to the catalog field or in other fields. These entries will be evaluated during the query.

The user can enter values from the created database list or enter new values.

From the properties window of a text field, select the row Catalog and then the entry **Database**. This catalog will not be configured.

The database catalog has no function in test mode.

Database catalogs are supplemented by an autocomplete function. The user is then given a list of suggestions based on existing entries when they enter three characters or more when searching, editing, or creating.

The rights system for the object type is taken into account, clauses on referenced folder or register types are not taken into account. You can stop the rights system from being taken into account via the following entry on the **Database catalog** tab:

```
IGNORESECURITYCHECK=TRUE
```

Other fields that have already been filled out are also considered when determining the suggestions.

You can stop already filled fields from being taken into account during creation and editing by adding the following entry to the **Database catalog** tab:

```
IGNOREOTHERFIELDINEDIT=TRUE
```

Autocomplete can be switched off for all database catalogs via an entry in the `as.cfg` file, located in the `\etc` directory of the data directory:

```
[SYSTEM]
```

```
USETYPEAHEADONDBCATALOG=0
```

Fields with a database catalog need a database index for the corresponding table column for auto-complete.

Structure tree

The structure tree catalog corresponds to a hierarchy catalog. From the structure tree catalog, the values selected by the user are transferred to the indexing form using the default abbreviations.

For example:

A link has the levels:

Year - Editor - Month – Process type

As an entry, the user selects:

1998 - Paul Werner - October – Family law

As an abbreviation, the following is entered into the catalog field:

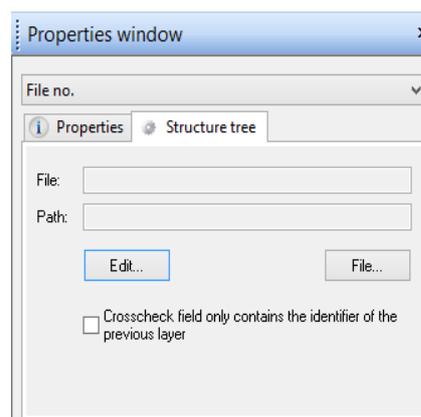
98-PW-10-F

The values from the structure tree can be edited by the user in the query, for example a placeholder '*' can be inserted. Users can enter values to the catalog field independent of the catalog. If the values are not contained in the list, the user receives an error message.

The values are saved to a file. Please note that all users can access the file from enaio® client and also from enaio® Document Storage.

How to create a structure tree catalog:

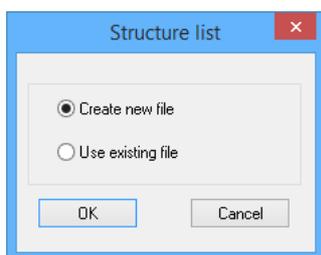
1. From the properties window of a text field, select the row **Catalog** and then the entry **Structure tree**.
2. Open the **Structure tree** tab.



If a structure file tree already exists, you can open it via the **File** button.

You can also specify on the tab if the content of all levels or only the last level is transferred.

3. Click the **Edit** button to create a new file.
The window Structure list will be opened.



4. Select one of the following options:

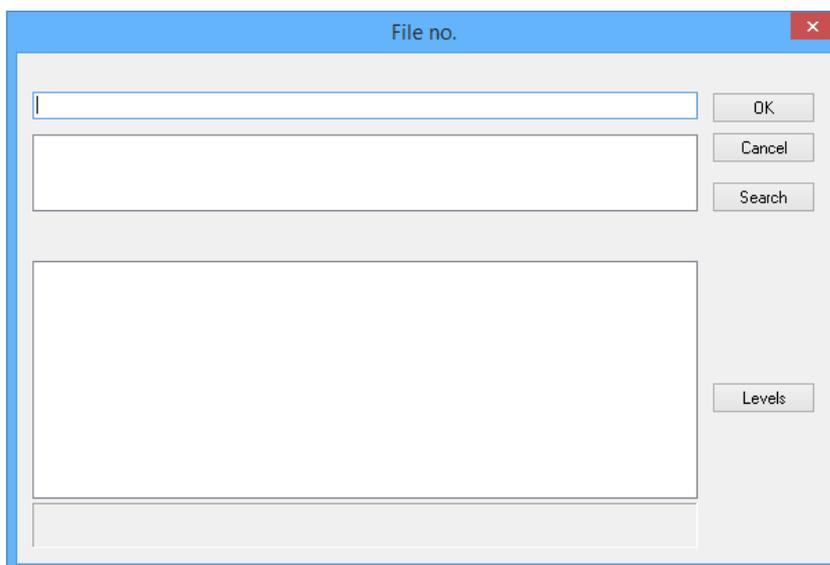
§Create new: Name the file. Save the file in a directory which is accessible to all users.

The editing dialog will be opened.

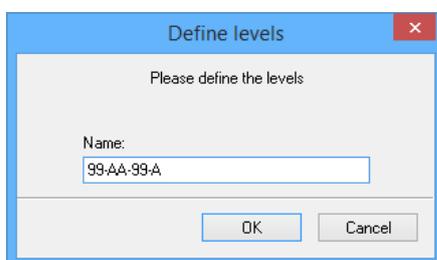
§Use existing file: Open a file. Definitions of structure trees are saved in files which have the extension 'dat'. You cannot save the file under a different name. Changes to this file also apply to all other structure tree catalogs which are defined for this file.

You can edit the structure tree.

5. Define the levels in the empty editing dialog:



6. Click the **Levels** button.
The window **Define levels** will open.



A level is defined by using place holders to specify the structure and the number of digits for the abbreviations.

As a placeholder you use:

§a letter, if a letter or digit is specified at the position for the abbreviation.

§a digit, if a digit is specified at the position for the abbreviation.

The level definition is separated by a separator. Use the character '@' if the separator should not be entered into the catalog field. The separators '-', '/', ':' will be entered into the catalog field. Separators can be replaced by any other character after the level definition.

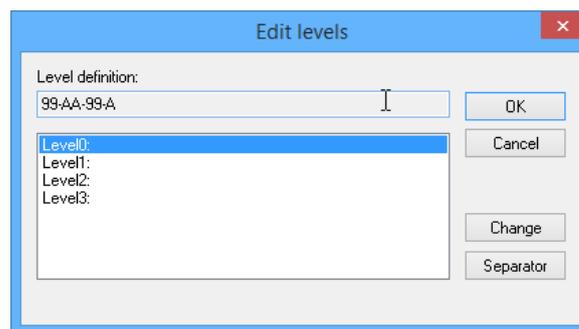
The level definition for the above example is as follows: 99-AA-99-A

Confirm the level definition with **OK**.

7. Assign names to the levels:

§Click the **Levels** button in the editing catalog.

§The window Edit levels will open. Here you can specify the names for the levels. The names are for the purposes of clarity.



§Select a level, click the button **Change** and enter a new name for the level.



In the Edit levels window you can also use the button **Separator** to enter a new separator for the selected level.

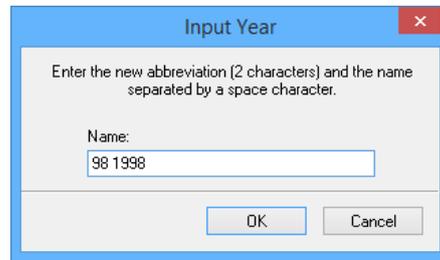
In the example the levels have the names: Year, Editor, Month, Process type.

The last level can be deleted but no new levels can be added and you cannot change the level definition.

§Confirm the level name with **OK**.

8. Create a list with values and abbreviations for the first level:

§Click on the **New** button. In the Entry window, enter an abbreviation for the defaulted definition, a space, and a value.

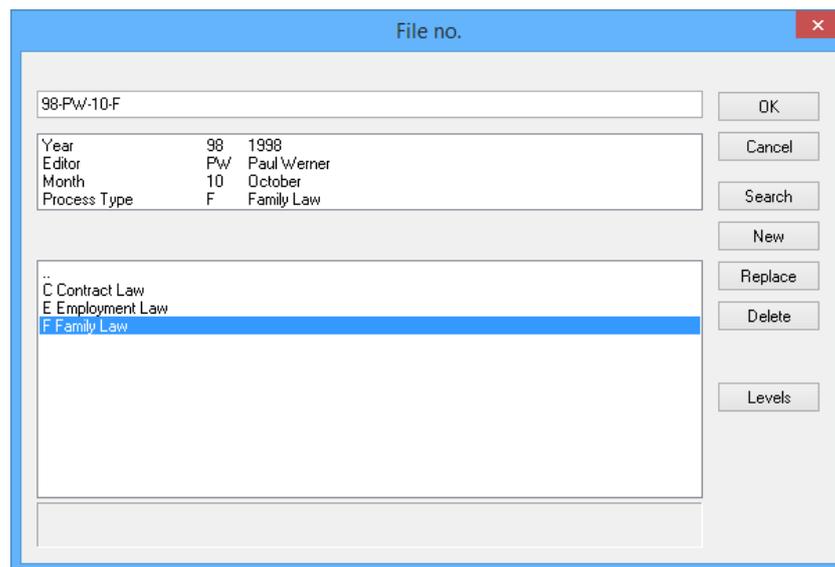


In the example the entry for the level 'Year' with the definition '99' is:
98 1998

§ Confirm the entry with **OK**.

Entries can be selected and deleted. Entries are edited by using the button **New** in the window Input and reentering an abbreviation with the edited value.

9. Create a list with the values and abbreviations for the next level:



§ Double-click an entry on the first level in the lower section of the editing dialog. The definition section of the next level for the entry of the first level will be opened.

§ Click on the **New** button. Enter an abbreviation in the Entry window according to the default definition for the level, a space, and a value and confirm the entry with **OK**.

In the example the entry is for the second level 'Editor' with the definition 'AA': PW Paul Werner

To get to the definition section of the next level, double-click an entry from the current level.

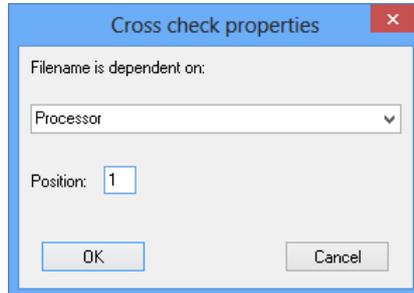
To get to the definition section of the previous level, double-click the first row which has a line of dots.

10. Save the structure tree by clicking the **OK** button in the editing dialog. It will be saved under the name of your choice.

As soon as you have saved the structure tree, the button **Copy** is available to you. With this button you can copy an entry, along with all subordinate entries, and insert it at a different location. As a target you enter the abbreviation of the target level followed by a separator. If the specified abbreviation does not exist at the target you can create it.

You can assign the property **Controlled by cross check** to another text field and then refer to the catalog field for the structure tree. Then the entry which stands for this abbreviation will automatically be inserted into the field. The level entries are separated by the character '\\'.

If you select the property **Controlled by cross check** for a field, the dialog **Cross check properties** will be opened.



You then select the structure tree from the list and assign position '1' to it.

The `oxlist.dll` library provides an API interface to convert ASCII text files into structure tree files and vice versa. This allows data for structure tree catalogs to be created dynamically, to update and to align. Further information can be found in the appendix of the 'enaio® client – Programming Reference' manual.

Exporting and Importing Catalog Data

You can export and import the data associated with list, tree and hierarchy catalogs.

Thus, data can be moved between test systems and deployed systems. Data for structure trees are already available as files and can be exchanged in the same way.

Data is exported from the corresponding catalog tab. The **Export catalog data** and **Import catalog data** functions can be found in the context menu.

If you want to import catalog data, you need a catalog file exported as such, or an XML file that corresponds to the schema of an exported catalog file.

Upon request, we can provide you with an XSD schema.

Properties like 'Sort' and 'Multi selection' for list catalogs and the separator for a hierarchy catalog are not exported or imported. Errors can occur for icon catalogs if imported catalog data contain user-defined icons which are not present on the destination system.

AddOns for Text Fields

AddOns allow you to integrate additional functions into enaio®.

The following AddOns belong to the enaio® delivery package:

§ Quickfinder AddOn

A Quickfinder field on an indexing dialog is linked to a field of a different folder type or document type and can only be indexed with the data which have been entered there.

§ WWW AddOn

The user can open a URL which has been entered into the index data field in the standard web browser, or send an e-mail message using the standard mailing program, to an e-mail address specified in a field.

§ Date AddOn

The user can select a date using the calendar.

§ Conversion AddOn

The add-on multiplies the number entered in the field with a configurable factor and transfers it to a different field.

§ Filing Plan AddOn

The filing plan AddOn attaches a number, a '/' character, and the current year to the indexing of a field or parts of the indexing of other fields in a linked field. The number is incremented by '1' each time it is attached.

§ Counter AddOn

The counter AddOn enters, similar to the filing plan AddOn, the indexing or parts of the indexing of other fields, a counter and the date or parts of the date in the AddOn field.

§ ID AddOn

The ID add-on creates a unique and consecutive number for the index data field.

§ User AddOn

The user add-on opens a list of all OS users and user groups.

§ Rights group AddOn

The rights group add-on opens a dialog, from which OS users and user groups can be selected. Access rights can be controlled using this add-on and corresponding logical expressions.

§ Application add-on

The user can start any program.

§ Query AddOn

The user can start a query.

§ Address AddOn

The user can transfer the index data of another object into fields of the indexing form.

§ Catalog AddOn

The catalog add-on can be configured as a list, tree, or hierarchy catalog. A multiple selection is possible for every catalog type. Unlike these catalogs, it is easier to address the catalog AddOn using scripts as it has a COM interface. The catalog data are saved in an XML file.

§ Database AddOn

The database add-on queries a database table of an external database and transfers the data into the index data field and connected fields.

Via the optional VB Script AddOn VB scripts can be executed. VB-Scripts are customized for your individual needs by OPTIMAL SYSTEMS GmbH or created by you with the VBScript editor.

According to the default settings, the AddOns do not write in fields with the 'read-only' property or in supervisor fields. If you want to change this, enter the following as an entry in the configuration field: CANLOCK=0

The value CANLOCK=1 will switch back to the default setting.

AddOns on forms which are opened as read-only are only executed if you have specified CANLOCK=0 as an entry in the configuration field.

Using the CANLOCK=2 setting, add-ons are not run on write-protected forms. On non-write-protected forms, the add-ons are run even when fields have the property 'write-protected' or 'supervisor field'.

AddOns verify the entries in the connected fields before saving. This can cause, depending on the AddOn, user entries in AddOn fields to be marked as incorrect or automatically corrected. The verification can be turned off with the entry SKIPVERIFY=1 in the configuration field.

AddOns with the property 'Execute automatically' are not verified.

Quickfinder AddOn

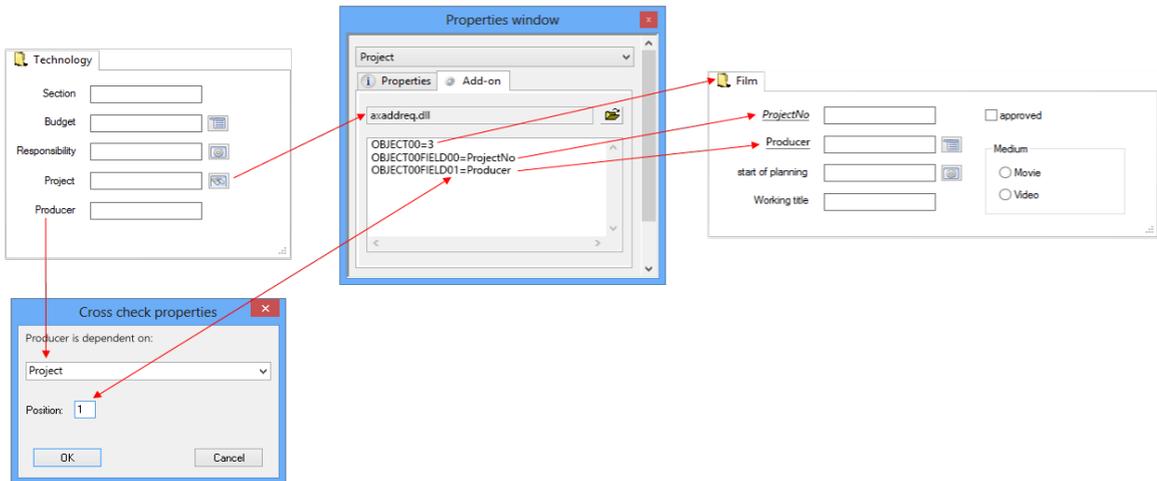
With the 'Quickfinder' AddOn, the Quickfinder field in the indexing dialog will be linked to a dialog element of a different object type. You can link multiple dialog elements of the indexing dialog using the Quickfinder field. Here you only have to select one dialog element as a Quickfinder field. The other dialog elements get the property Controlled by cross check. The connection can be set up for any desired folder, register or document type. Multiple object types can be specified. Then the user selects the desired type.

In the Quickfinder field, the user takes the value from the indexing of an object of the connected type. This object is queried by the user using the embedded search form. The corresponding index data will be applied to dialog elements with the Controlled by Cross Check property. The Quickfinder field can be edited independently of the Quickfinder.

Example:

The field 'Project' of the object 'Technics' is linked to the object 'Film' via the Quickfinder AddOn. The object 'Film' has ID '3' (OBJEKT00=3).

The Quickfinder field 'Technics/Project' transfers the data from the field 'Film/ProjectNo:' (OBJEKT00FELD00'). The field 'Technics/Producer' is dependent on the Quickfinder field 'Technics/Project' via the crosscheck property, it has position '1' and contains data from the field 'Film/Producer' (OBJEKT00FELD01').



Connections between indexing forms with the 'Quickfinder' AddOn

If the Quickfinder field is connected to a folder field then the user can quickly create a new folder via the Quickfinder. This is not possible for registers and documents.

No AddOns can be executed when creating a new folder using the Quickfinder.

New registers and documents cannot be created if they are linked.

If a user changes the indexing of the connected object at a later date, the changes will not be transferred to the Quickfinder field.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddreq.dll` via the file selection dialog. It is found in the OS directory `\client32`.

The entries in the configuration field have the following structure:

```

OBJEKT00=ObjectTypeID
OBJEKT01=%internerNameObjektyp%
...
OBJEKT00FELD00=NameFieldPosition0
OBJEKT00FELD01=NameFeldPosition1
...
OBJEKT01FELD00=%internalNameFieldPosition0%
OBJEKT01FELD01=%internerNameFeldPosition1%
...
    
```

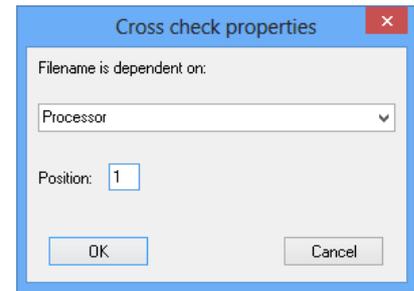
Firstly, you list the archive objects which the user can select from, with consecutive numbers. The ID of the object type is specified as the first value in the properties dialog in the row Object type. Instead of this value you can specify the internal name of the object type, adding the character '%' in front of and behind the name.

In the second part you specify the fields, in ascending order, from which data should be transferred. As a name, you use the name of the field or the internal name. Internal names are selected by placing a '%' in front of and behind the name.

A transfer from table fields is not possible. Always use the internal name when you create language versions. The Quickfinder AddOn causes errors if the language used in the configuration is different to the one used in the client.

The data from 'OBJEKTnFELD00' will be used in the Quickfinder field. The entries from 'OBJEKTnFELD01' on, which have the property Controlled by cross check with the position number '1' will be used.

In the dialog **Cross check properties**, select the Quickfinder field from the list and enter the position for FELD01 '1'.



For every further field 'OBJEKTnFELDn' via the property Controlled by cross check, enter the position of the field which should use the data.

Base parameters can be applied in addition to the index data. Use the following names for base parameters:

@@OBJECTTYPE@@	The object type
@@OBJECTID@@	The unique ID of the object
@@OBJECTNAME@@	The name of the object type
@@CREATOR@@	The creator
@@CREATIONTIME@@	The creation date
@@TIMESTAMP@@	The creation date as a timestamp
@@MODIFYUSER@@	The name of the user who made the last change
@@MODIFYTIME@@	The timestamp of the last change
@@ARCHIVIST@@	The archivist of documents
@@ARCHIVETIME@@	The date of archiving of documents

Using an extra entry, the Quickfinder add-on security system can be configured:

EXTRA00=0 (Enforce security system – default setting)

EXTRA00=1 (Do not enforce the security system, allow queries, creation, and editing)

EXTRA00=2 (Do not enforce the security system, allow queries, do not allow creation or editing)

Another EXTRA entry allows you to specify whether to have present content in Quickfinder fields be deleted when data is transferred, provided that the fields being connected with the Quickfinder fields do not contain any data.

EXTRA01=0 (Do not delete – default setting)

EXTRA01=1 (Delete)

When saving the indexing data, the Quickfinder executes a query and verifies the data transfer. If a user changed applied data, the changes are undone through the verification. The verification can be turned off with the entry SKIPVERIFY=1. The verification leads to delays and can therefore be turned off.

WWW AddOn

With the WWW AddOn, a user can open the URL entered into the index data field with a standard web browser, or transfer the e-mail address specified in the index data field to an e-mail form of the standard e-mail program.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddwww.dll` from the file selection dialog. It is found in the OS directory `\client32`.

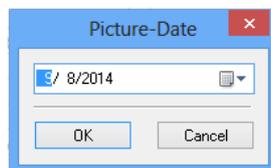
No further configuration is required.

If the AddOn is placed on a form which is opened as read-only, it is only executed if you have specified `CANLOCK=0` as the last entry in the configuration field.

Date AddOn

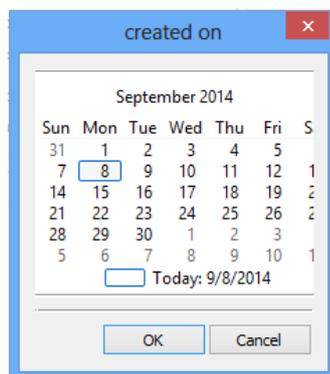
The date AddOn adds a calendar. The user can use the calendar to choose a date to enter in the field.

If the user clicks on the **AddOn** button, an editable field will be opened in which the user can enter a date. The field is preset with the current date.



The user can open a calendar using the associated arrow button and transfer a date from it to the editable field.

Insert the entry `EXTRA00=CALENDAR` into the configuration field in order to directly open the calendar and to transfer the selected date directly to the index data field.



If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddate.dll` from the file selection dialog. It is found in the OS directory `\client32`.

No further configuration is required.

Conversion AddOn

The conversion AddOn is especially useful for currency conversion. The AddOn multiplies the number entered in the field with a configurable factor and transfers it to a different field. The standard factor is the DM-euro conversion factor 0.51129. You can enter a different factor.

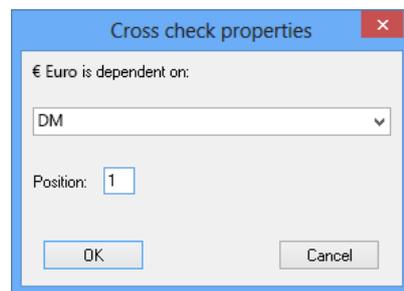
The conversion factor will apply to all uses of the AddOn in the system.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddeur.dll` via the file selection dialog. It is found in the OS directory `\client32`.

For the field to which the converted number is entered, configure the property **Controlled by cross check**.

In the dialog **Cross check properties** from the list select the conversion AddOn field and enter the position '1'.



The conversion AddOn uses the standard conversion factor 0.51129. If you want to use a different conversion factor, enter the factor in the file `as.cfg`. The file is found in the `\etc` directory of the data directory. You can edit the file with any editor.

Fill out the file with entries of following structure:

```
[ASADDEUR]
Factor=Value
```

Example:

```
[ASADDEUR]
Faktor=1.9558
```

Filing Plan AddOn

The filing plan add-on enters the indexing parts of the indexing of other fields in the linked field and adds a counter, a '/'-character, and the current year. The counter is always increased by the value '1' but it can be reset at certain points.

Which indexing or part of the indexing of other fields should be used is determined in the configuration field on the **AddOn** tab.

Index data from other folder, register, or document types cannot be applied.

You assign the filing plan AddOn to a register indexing field or a document index data field.

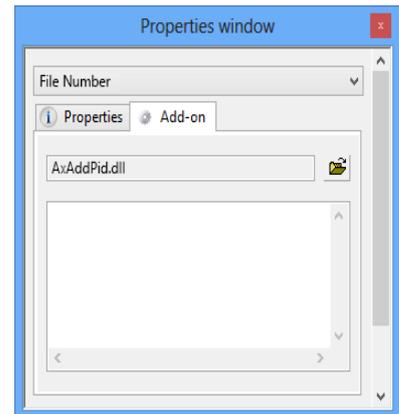
On the search forms the filing plan AddOn has the function of a database catalog.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddpid.dll` via the file selection dialog. It is found in the OS directory `\client32`.

In the configuration field, enter the references to the fields from which data should be transferred.

You can select from two different types of indexing. Each format has a different structure.



Format with Reference to Folder/Register Indexing

Example: Customer-Order-5/02

The entries in the configuration field have the following structure:

```
EXTRA00=Foldername@Fieldname,Startposition,Characternumber
EXTRA01=Registername@Fieldname,Startposition,Characternumber
```

In the first row, you specify the reference to the folder in which the register or document should be created. You specify the field name of the field whose indexing should be transferred, a start position and the number of characters. The start position is the first character which is transferred; the first character has the position '0'. The number of characters specifies how many characters following the start position should be transferred.

Example: `EXTRA00=FolderName@FieldName,3,3`

In the example, the fourth character is the first one to be transferred; it has the start position '3'. Altogether there are three characters transferred, the fourth, fifth and the sixth character.

If you specify an internal name insert a '%' in front and behind the internal name.

If you do not specify values for the start position, the standard value '0' will be used for the start position and '3' for the number of characters will be used.

The entire indexing will be used if you specify '0' for the start position and '0' for the number of characters.

If you do not transfer characters from the folder indexing, enter the following:
`EXTRA00=unused`

In the second row you specify the corresponding reference to the register field with the start position and the number of characters. If you do not transfer characters from the register indexing, enter the following: `EXTRA01=unused`

If you assign the filing plan AddOn to a register field, only the indexing of a corresponding folder field can be transferred. The second row must specify `EXTRA01=unused`. If the AddOn is assigned to a document field, it can only be executed when newly created, if the document is located in a register or if the second row specifies `EXTRA01=unused`.

If you want to regularly reset the counter, add the following entries:

```
EXTRAn=COUNTERTYPE='Type'
EXTRAn+1=INITIALVALUE='Beginningvalue'
```

For the type of counter enter one of the following:

- § 0 if the counter should not be reset,
- § 1 if the counter should be reset daily,
- § 2 if the counter should be reset monthly,
- § 3 if the counter should be reset yearly.

For the beginning value, specify a numeric value which the counter should be reset to.

Without counter entries the counter will not be reset.

The number of digits for the counter is entered in this way:

```
EXTRAn=COUNTERWIDTH='Digits'
```

If the object with the current counter is deleted, but still is in the trash can, the default for the counter will not use this number but the next higher number for the next object. You can change this behavior with the following entry:

```
Extran=Deleted=1
```

Then, when determining the counter number for a new object, the objects in the trash can will be ignored. However, if an object is recovered from the trash can it can happen that two objects have the same counter number.

The year format can be set with two or four digits:

```
EXTRAn=YEARFORMAT='Digits'
```

As 'digits' you should specify '2' or '4'.

You number the entries on the **AddOn** tab consecutively, beginning with EXTRA00.

Format with Reference to Document Indexing

This format has to be assigned to a document index data field. You must specify the reference to a folder field, a register field and a document field. The document has to be created in a register by the user.

Example: Cus-On-Ord-5/2000

The entries in the configuration field have the following structure:

```
EXTRA00=Foldertypename@Fieldname
EXTRA01=Registertypename@Fieldname
EXTRA02=Documenttypename@Fieldname
EXTRA03=CROSSCHECKFIELD=Startposition,Characternumber
```

In the first row you determine the reference to the folder where the document should be created. You specify the field name whose indexing data should be used. The first three characters from this field will be used.

In the second row you specify the reference to a register field. The first three characters from this field will be used.

In the third row you specify the reference to an index data field from the same indexing dialog. Assign the property **Controlled by cross check** to the field from which the characters should be used. Via the dialog **Cross check properties**, specify the reference to the AddOn field and the number '1' as position.

For the document type field you can additionally specify a start position and a character number. This entry is optional, if you don't enter anything, the first three characters will be used. The start position for the first character is '0'.

This is followed by the entries for counter type, beginning value and the ignoring of deleted objects. Without making an entry the counter will not be reset.

If you specify an internal name insert a '%' in front and behind the internal name.

Counter AddOn

With the Counter AddOn, in the AddOn field you enter, similar to the filing plan AddOn, indexing or parts of the indexing of other fields, a counter, and the date or parts of the date.

The filing plan AddOn can only be assigned to register and document index data fields, but the counter AddOn can be used for folder index data fields. Next to the year you can also enter the month and the day, and there are more configurable options.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddcnt.dll` from the file selection dialog. It is found in the OS directory `\client32`.

For the AddOn, you specify a type of counter, the indexing reference, and the format in an extra entry:

```
EXTRAn=Type=Property
```

You number the extra entries on the **AddOn** tab consecutively, beginning with EXTRA00. The type order is irrelevant.

Counters

You configure the counter in the same way as for filing plan AddOn, via the properties COUNTERTYPE, COUNTERWIDTH, and INITIALVALUE.

Additionally, the logic of the counter can be configured. Without further information, the counter will count higher independent of the other components and is reset via the property COUNTERTYPE.

If you also set up the property EXTRAn=USEDBCOUNTER=1 the logic of the counter will work differently. The database will then be searched for an entry which exactly matches and the counter will be increased by exactly '1' relative to the entry found. If this is the case the reset property COUNTERTYPE will not be evaluated.

If the object with the active number is deleted, the counter number for the object will not be newly distributed but the next higher number will be used. In contrast to the filing plan AddOn, this behavior cannot be changed.

Indexing reference

Fields on the same dialog from which the indexing or parts of the indexing should be adapted should have the property Controlled by cross check. Via the dialog **Cross check properties** you specify the reference to the AddOn field and the position '1' for the first field. For the following fields you increase the position by one.

Via the configuration field of the Counter AddOn, use the extra entries to specify which parts of the indexing should be adapted:

```
EXTRAn=Fieldn=Startposition,Characternumber
```

You number the field entries starting with Field0, according to the order of the positions specified in the configuration of the cross check property. With Field0 you refer to the field with position 1.

The start position is the first character which should be adapted; the number of characters specifies how many characters following the start position should be adapted. If you do not specify values for the start position, the standard value '0' will be used for the start position and '3' for the number of characters will be used. For the entire indexing, specify '0' for the start position and the maximum field length for the number of characters.

In order to refer to the location data, meaning the indexing of the assigned folder or register, you use entries with the following structure:

```
EXTRAn=OFieldn=Objecttype@Fieldname,Startposition,Characternumber
```

The location fields are selected with a leading letter 'O'. Object type and field name are specified and the start position as well as the character number.

If you specify an internal name insert a '%' in front and behind the internal name.

The location field entries are numbered beginning with OField0.

If for example a document has multiple register locations you can create multiple references.

Example:

```
EXTRA02=OFIELD1=Register1@Editor,0,4
EXTRA03=OFIELD1=Register2@Responsible,0,4
```

Depending on the location the entry 'OFIELD1' will be filled out with register1 data or with register2 data.

A reference to a register is only possible for objects which are stored directly in a register.

Format

Then you specify exactly how the counter, the configurable field entries, the date, and the various strings should be arranged in the indexing field.

```
EXTRAn=FORMAT=...<Counter>...<Fieldn>...<OFieldn>...<DateFormat>...
```

... stands for any character sequences that you can enter into the field as separators or as constant text.

Strings used here may not contain the characters '>' or '<'.

<Counter> is the configured counter, <Fieldn> stands for the configured indexing reference of the same form, <OFieldn> for the location reference.

If a field which you are referring to is not indexed, the digit will stay empty. If you assign the property OPTIONAL to the fields, the leading strings will be removed.

```
EXTRAn=OPTIONAL=<Fieldn>;<OFieldn>
```

The date for the is specified as follows:

§ <YYYY> for the current four-digit year

§ <YY> for the current two-digit year

§ <MM> for the current month

§ <DD> for the current day.

Example:

```
EXTRA00=COUNTERWIDTH=5
EXTRA01=COUNTERTYPE=2
EXTRA02=INITIALVALUE=1
EXTRA03=FIELD0=0,4
EXTRA04=OFIELD0=Register@Number,0,2
EXTRA05=OPTIONAL=OFIELD0
EXTRA06=FORMAT=Support <YY>.<MM>/<COUNTER> <FIELD0>-<OFIELD0>
```

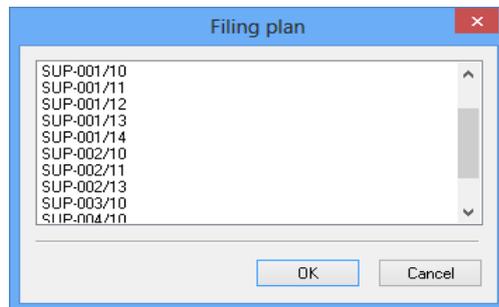
The counter has five digits (COUNTERWIDTH) and is reset monthly (COUNTERTYPE) to the value '1' (INITIALVALUE). Reference is made to fields on the same dialog and the first four digits are used (FIELD0=0,4). Reference is also made to the field 'Number' of the object type 'Register'. Here the first two characters are used.

The string 'Support,' the two-digit year, a dot as a separator, the current month, the separator '/', the counter, a space as a separator, the content of field 'Field0,' a minus sign, and the content of 'OField0' will be output.

If the field 'OField0' is empty the minus sign will be dropped.

Example: Support 03.05/00005 1910-ab

Similar to the filing plan AddOn, the **AddOn** button on the search form will open a dialog where all of the field's index data are listed.



ID AddOn

The ID AddOn creates a unique and consecutive number for the index data field. The number is entered into the field when the user clicks the **AddOn** button. It corresponds to the object ID.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file axaddidx.dll via the file selection dialog. It is found in the OS directory \client32.

The ID AddOn can only be used once within a given dialog.

No further configuration is required.

User AddOn

The user AddOn opens a list of all users and user groups created in enaio® administrator. The list contains the user name and the entries from field 'complete name' for all users. The user name or the entire name is transferred.

However, users can also enter any other values in the field.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddusr.dll` from the file selection dialog. It is found in the OS directory `\client32`.

The entries in the configuration field have the following structure:

```
EXTRA00=GROUPS,MULTISELECTION,FULLUSERNAME
EXTRA01=Value1,Value2,....
```

In the first row you specify whether the list should also contain all groups, whether a multiple selection is possible and whether the entire user name should be used instead of the login name. The intended values should be separated by commas.

In the second row you can specify optional additional values for the list. The values should be separated by commas.

Without additional entries the user list opens without multiple selection and with the transferred login name. If you require the user list and other values, enter 'EXTRA00=BENUTZER' in the first line.

The entries of the user AddOn cannot be sorted by the user. However, sorting by user name or user description can be defined with an extra entry in the configuration field. The following configuration parameters can be used for this:

SORTCOL1 - sorts alphabetically and ascending by user name.

SORTCOL2 - sorts alphabetically and ascending by full user name.

The extra entries have to be consecutively numbered beginning with '00'.

For a user list you can reduce the selection to those users who are members of at least one of the groups in which the user who opens the list has membership. To do so, the following entry must be added to the `as.cfg` file, located in the `\etc` directory of the data directory:

```
[CLIENT]
```

```
HideOthers=1
```

You can add another line to exclude single users from this function. These users will continue to see all other users:

```
Exclude=User1;User2
```

Rights group AddOn

The rights group AddOn opens a dialog, from which OS users and user groups can be selected. Via this AddOn and corresponding logical expressions which are set using enaio® administrator, access rights for the indexing can be controlled. For example, access to an object can be restricted to users of a certain group or to individually listed users if a field is indexed using this AddOn.

Details can be found in the Administrator manual.

However, users can also enter any other values in the field.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddusrgrp.dll` via the file selection dialog. It is found in the OS directory `\client32`.

Via the configuration entry, you specify whether users and groups, only users, only groups or listed groups or users should be listed.

The following entries are possible:

<code>EXTRAn=BOTH</code>	Default setting: all users and all groups
<code>EXTRAn=USER</code>	Only users
<code>EXTRAn=GROUPS</code>	Only groups
<code>EXTRAn=USER_EXCLUDE: name1;name2</code>	All users except the specified
<code>EXTRAn=GROUPS_EXCLUDE: group1;group2</code>	All groups except those specified.
<code>EXTRAn=USER_EXCLUSIVE: name1;name2</code>	Only the specified users.
<code>EXTRAn=USER_FILTER:Prefix</code>	Only users whose names begin with the specified prefix. Multiple prefixes are separated by a semicolon.
<code>EXTRAn=GROUPS_EXCLUSIVE: group1;group2</code>	Only the specified groups and their users. With <code>EXTRAn=USER</code> only the users of the groups, with <code>EXTRAn=GROUPS</code> only the groups
<code>EXTRAn=GROUPS_FILTER: Prefix</code>	Only the groups and their users whose names begin with the specified prefix. Multiple prefixes are separated by a semicolon.
<code>EXTRAn=SORTCOLn</code>	<code>SORTCOL1</code> - sorts groups and users in alphabetical ascending order by user name. <code>SORTCOL2</code> - sorts users and groups in alphabetical ascending order by user name. <code>SORTCOL3</code> - sorts users and groups in alphabetical ascending order by user description.
<code>EXTRAn=CANSHOWDETAILS</code>	Detailed information on users or groups can be shown via the context menu in enaio® client.

The extra entries have to be consecutively numbered beginning with '00'.

In the index data field, a user '(U)' has been attached and a group '(G)' has been attached. The entries are separated by a semicolon. The number of entries in the index data field is restricted by the specified field length. If the user specifies entries which exceed the field length they will receive an error message.

If you select 'User(u)' as the preset function for the field, it is preset with the user name which is also generated with the rights group AddOn.

Just like with the user AddOn you can reduce the selection for a user list to those users who are members of at least one of the groups in which the user who opens the list has membership. To do so, the following entry must be added to the `as.cfg` file, located in the `\etc` directory of the data directory:

[CLIENT]

HideOthers=1

You can add another line to the section in order that other individual users are included even if they do not share group membership with the current user:

Exclude=User1;User2

In a group list, the **HideOthers** entry is used to show only the groups where the user is a member. With an 'Exclude' entry further groups can be added here as well.

Application add-on

The application AddOn allows the user to start an application.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddexe.dll` from the file selection dialog. It is found in the OS directory `\client32`.

The entries in the configuration field have the following structure:

```
EXTRA00=Path\Application|WAIT
```

You specify the application with its full path. Be sure that all users can access the application at the specified path.

If you specify the optional parameter 'WAIT', the client waits until the user has closed the program again.

If the AddOn is placed on a form which is opened as read-only, it is only executed if you have specified `CANLOCK=0` as the last entry in the configuration field.

Query AddOn

The query AddOn enables the user to execute a restricted query. If you click the **AddOn** button the query is executed and a hit list will open. The list shows documents which contain an index data field which has the exact name as specified in the AddOn field and which are indexed in this field with the entry in the AddOn field. The user can open the documents from the hit list.

For the query AddOn you need other document types with fields which have the same name as the field which you assign the query AddOn to. Via the configuration field you specify a document type to be used for the search.

Select the file `axaddbln.dll` via the file selection dialog. It is found in the OS directory `\client32`.

The entries in the configuration field have the following structure:

```
EXTRA00=Foldertype@Documenttypename
```

You specify the name of the folder type and then the name of the document type, separated by a '@'.

In the configuration, no internal names can be used. The AddOn cannot be used on page controls. For this query, no placeholders can be specified.

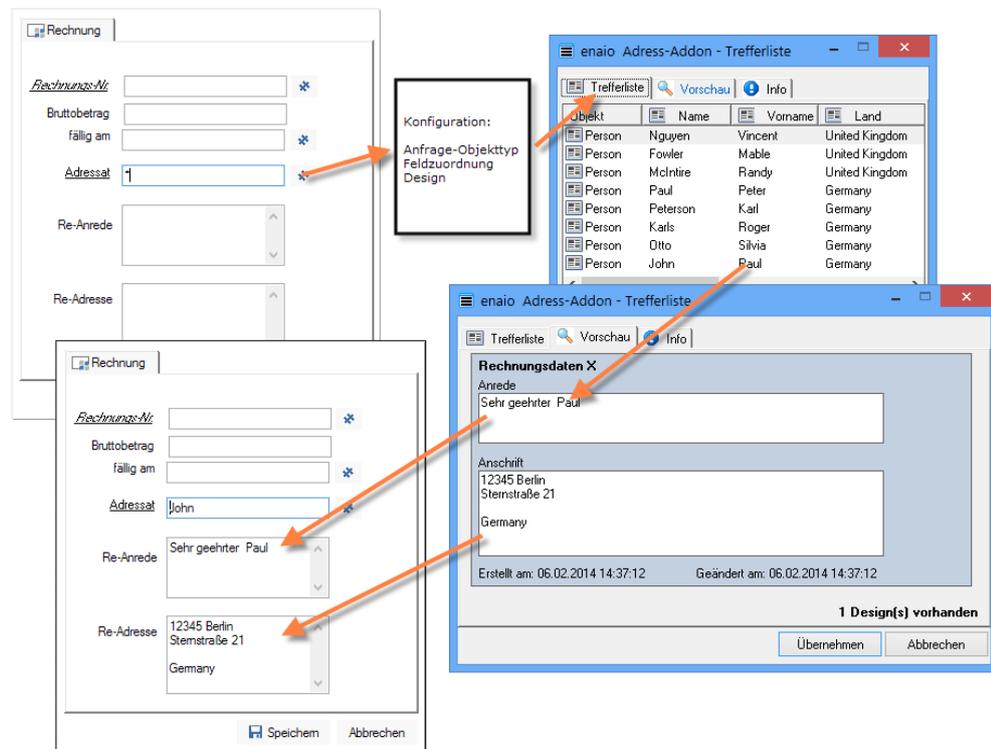
Address AddOn

The address AddOn transfers index data of a queried object into index data fields which are linked to the AddOn field through the property 'Controlled by cross check'.

Process:

The AddOn performs a query for the configured query object type with the data in the AddOn field and the crosscheck fields. The results are displayed in a hit list. Hits can be displayed in a preview. If more than one design templates are available, the user can select one and transfer the data in the corresponding format.

Schema:



The configuration is not performed in the configuration field in enaio® editor but in the wizard which is launched in enaio® client. To access the wizard a user requires the system role 'EDITOR: Start.'

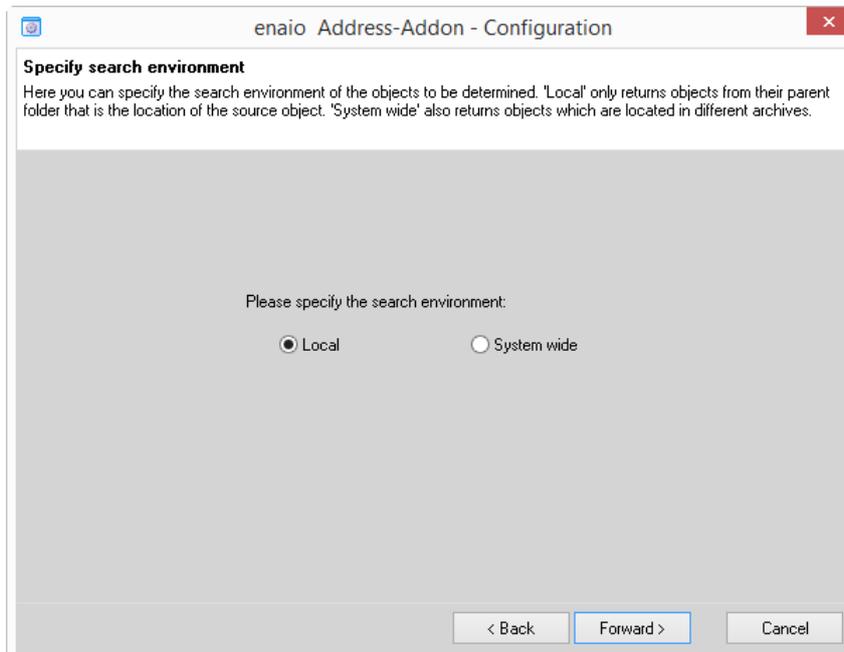
Assign the address add-on to a text field. Select the file `axaddaddress.dll` via the file selection dialog. It is found in the OS directory `\client32`.

For all other fields into which data are transferred you can specify the property 'Controlled by cross check' depending on the AddOn field and with a consecutive position number.

The address AddOn cannot be used if the object type names or field names contain a semicolon.

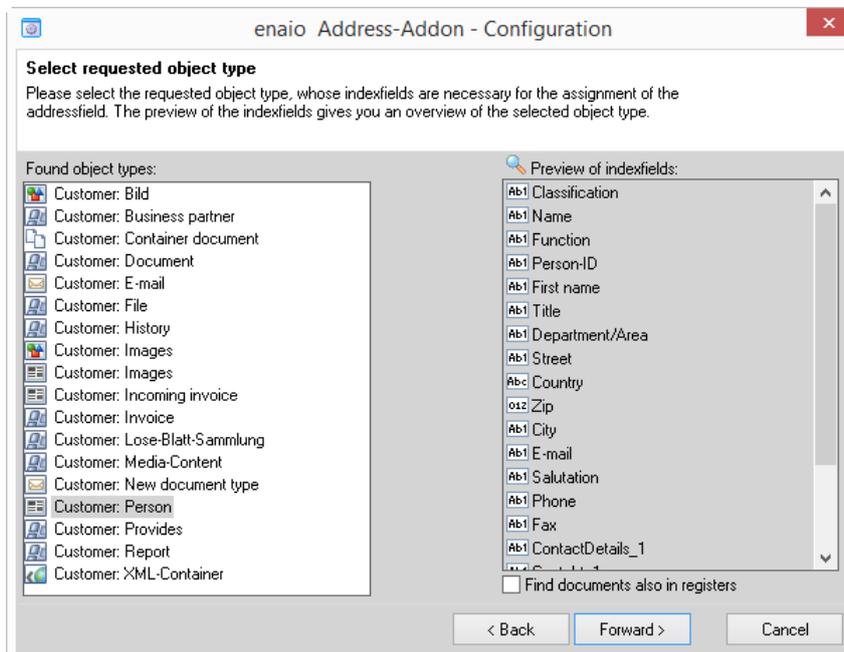
Configuration

Right-click the **AddOn** button on the form to launch the wizard in enaio® client. After a standard dialog you can specify the query range.



Specify, if you want to transfer location-related data of a query object or system-wide data of any query object.

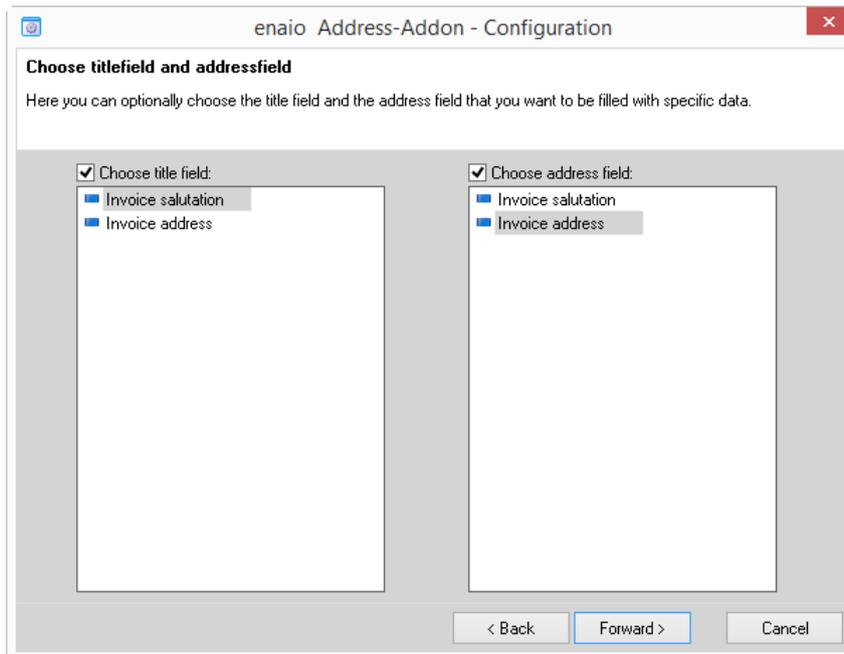
Select the query object in the following dialog.



Objects referring to their location or system-wide are listed on the left side. If you select an object, the object fields are listed on the right side.

Specify for all register and document types if only the topmost level in the folder is queried for objects or if all levels are queried.

In the following dialog you can select a Salutation and an Address field.

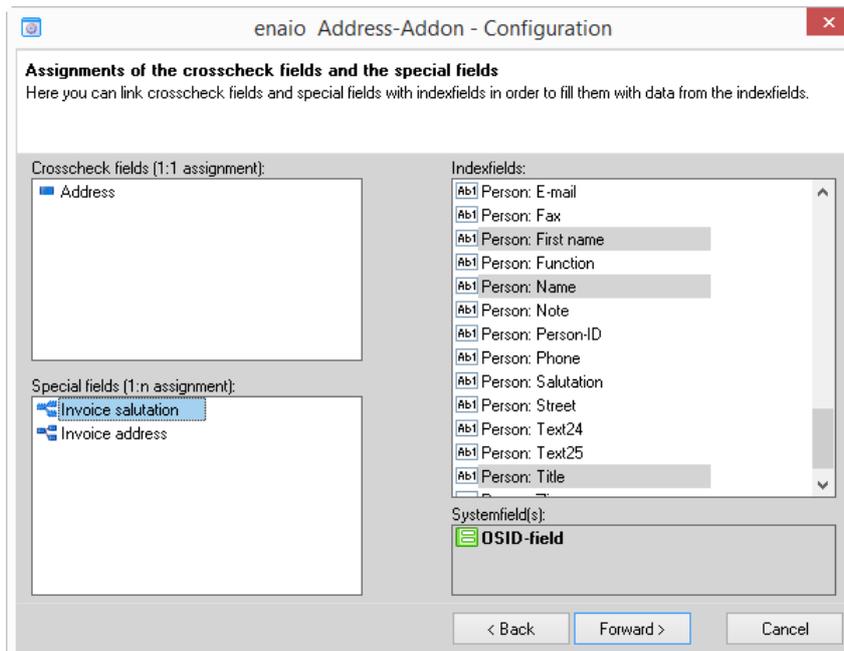


Salutation and address field are crosscheck fields to which you can map fields of the query object type. Data of multiple fields of a query object can be transferred into these fields. You can specify simple formatting with design templates.

Salutation and address fields are optional. The content is not analyzed when querying a object.

Only one field of the query object type can be assigned to all other crosscheck fields. This mapping is valid for queries and data transfer.

You can assign fields of the query object type to the AddOn field and all crosscheck fields in the following dialog.



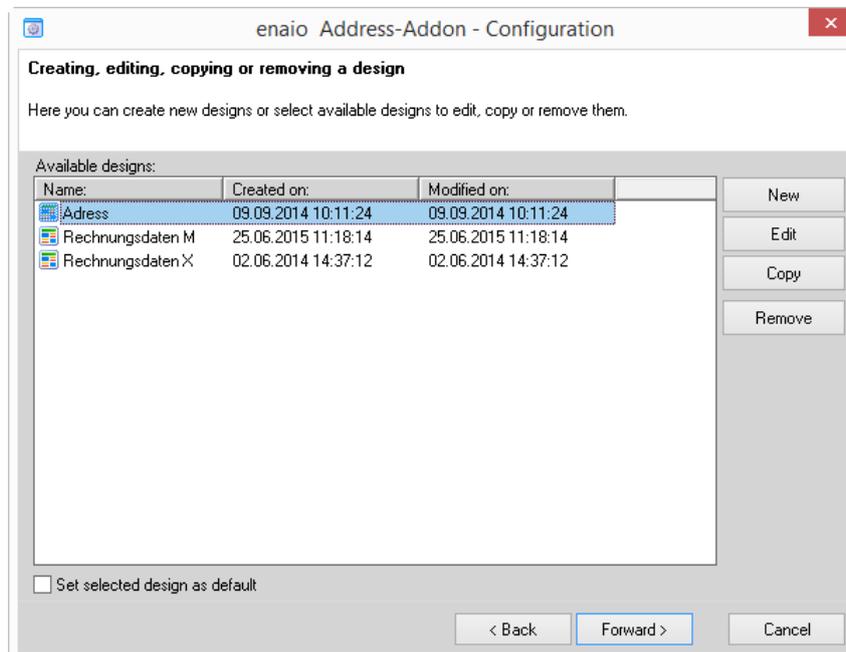
The AddOn field and all crosscheck fields which were not specified as salutation or address fields are listed in the Crosscheck fields area.

You can select a field and then a field of the query object type in the Index field area to create a mapping. A mapping with the OSID system field is also possible. Then a query for the ID of the query object is performed with the content of the AddOn or crosscheck field. More than one index field can be mapped for the special fields 'Salutation' and 'Address'.

Each field requires to be mapped to a field of the query object type or to the OSID.

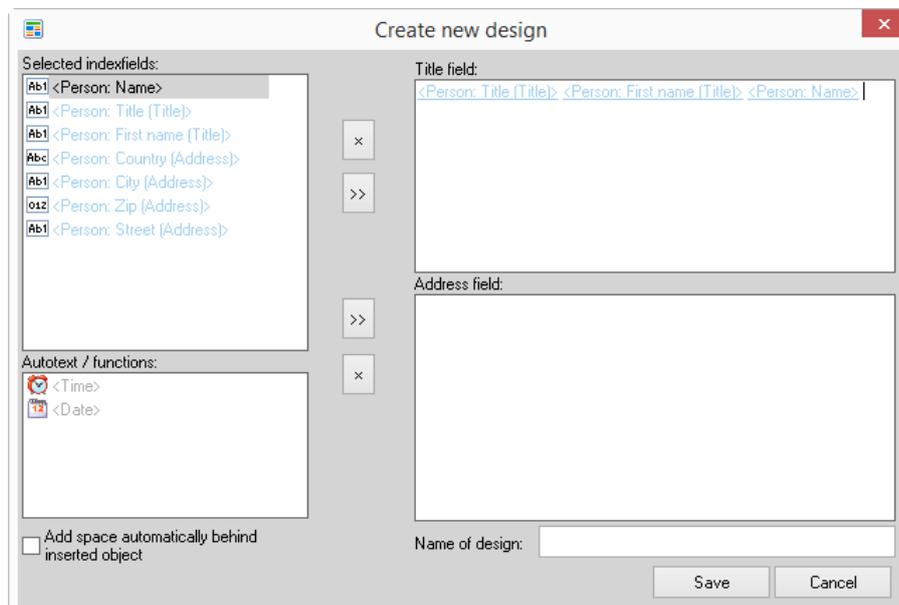
In the following dialog you can specify a design for the salutation and address field.

Those designs which have already been created are listed and can be edited.



You can define multiple designs and let the user select which one he wants to use. In a preview the user can decide which design will be used.

The **New** button is for the creation of a new design:



You can find all assigned fields of the query object type in the Selected index fields area. The possible mapping of salutation and address field is flagged. The AddOn field and other crosscheck fields can be mapped in any way using the arrow buttons.

You can enter any text, spaces, line breaks, and functions for the current date and the current time in the Salutation field and Address field area.

Enter a design name and save the data.

The new design will be displayed in the list of existing designs. You can create and edit more designs.

Click **Next** and **Finish**.

The configuration is saved as a file in the `\server\etc` directory.

If you are working with multiple servers, you have to copy the configuration file into the corresponding server directory.

Catalog AddOn

The catalog AddOn can be configured as a list, tree, or hierarchy catalog. Unlike these catalogs, the catalog AddOn has many other functions as well as a COM interface and can therefore be simply addressed with scripts.

It is possible to allow multiple selection for the user as a tree catalog or a hierarchy catalog.

Assign the catalog add-on to a text field. Select the file `axaddxmltree.dll` via the file selection dialog. It is found in the directory `\client32`.

Catalog data are saved in an XML file in the directory `...\server\etc`. The file is named 'axaddxmltree', followed by the hexadecimal field GUID. This field GUID can be found in the object definition file.

You can optionally specify a catalog file via the configuration field:

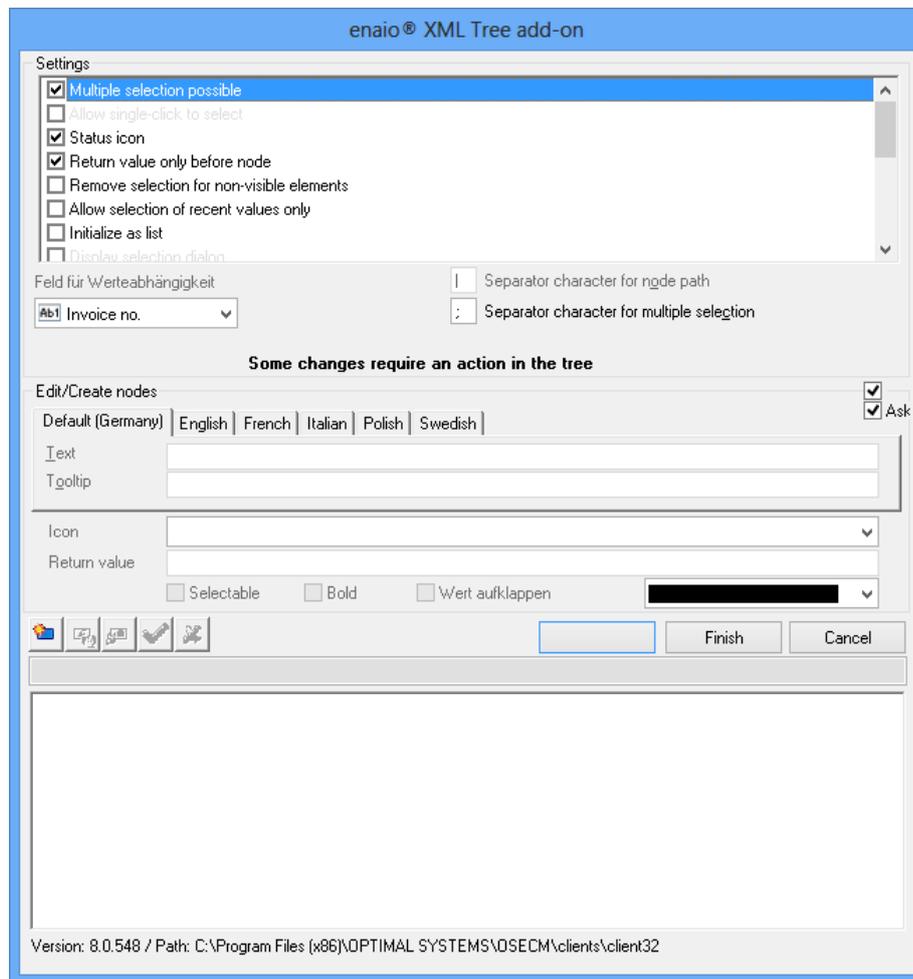
EXTRA00=bezeichnung.xml

The catalog file is created during configuration and saved in the directory `...\server\etc`. Catalog files can be assigned several catalog AddOn fields.

Configuration is performed using enaio® client and requires the system role 'Editor: Start.'

Please make sure that the data are not edited by more than one user at the same time.

Open the configuration dialog by right-clicking the **AddOn** button in enaio® client.



The dialog is divided into two areas: the settings area at the top which is used to set the catalog properties and the area at the bottom, used to create and edit catalog entries.

Settings

The following settings are possible for a catalog:

§ Multi-selection

Users can select several values from the catalog and transfer them into the field. Enter a separator for a multiple selection - a semi colon is set as default.

§ Allow single-click to select

Users transfer a value from the catalog with a single click on an entry. Default is double-click.

§ Status icon

Catalog values can be flagged with dot status icons in the catalog. Selected values are flagged with a green dot, not selected with a red one. A yellow dot is used to flag values with the property 'not selectable'.

If no status icon is displayed, all values are shown with a check box for multiple selection.

§ Return value only from the node

Only the value of the current level is transferred into the field.

A catalog with this property is equal to a tree catalog. Without this property, the catalog is equal to a hierarchy catalog. Enter a separator for the node path.

This setting has no effect on list catalogs.

§ Remove selection for non-visible elements

When using multiple selection, some values in nested catalogs may not be visible as the nodes of the values can be collapsed.

If you select this option, invisible values are not transferred for the field.

§ Allow selection of recent values only

Here you can specify if the user can only select the value of the last level or if he can select and transfer values from any other level.

A catalog with this property and the property 'Return value only from the node' is equal to a tree catalog.

A catalog with this property and without the property 'Return value only from the node' is equal to a hierarchy catalog.

A catalog without this property and without the property 'Return value only from the node' is equal to a hierarchy catalog with the property 'Allow intermediate levels'.

In addition you can specify for every single value if it can be transferred by the user or not.

§ Initialize as list

If you have already created elements with nodes, you can create a linear list via this option. When enabling the option, you can choose whether sub-elements are deleted or whether the sub-elements are appended to the linear list.

You can disable the option again after linearization. If you do not disable the option, the list will be shown in enaio® client for display only. You will not be able to transfer elements.

§ Show as selection dialog

List catalogs with multiple selection can be displayed as a selection dialog. Users can then transmit individual values in a selection range from the list of all values. This means that long lists can be more clearly arranged.

§ Verify entries and show message box

If a user enters a value independently of the catalog that is not included in the catalog, a notification is displayed when the **AddOn** button is clicked.

§ Validate when saving data sheet

If a user enters a value independently of the catalog that is not included in the catalog, a notification is displayed when the **AddOn** button is clicked.

The index data sheet stays open when saving and the value is not saved.

§ Sort values while loading

Entries in a node are automatically sorted alphabetically in the display in enaio® client.

§ Nodes have group limitations

The display of the values depends on which group the user is assigned to.

§ Nodes depend on the value of a field

The display of values depends on the value of another field. If the field is left empty, all values will be shown.

§ For value-dependent list: field value validation

If the display of the values is dependent on the value of another field, a notice is shown for an empty field.

§ Initialize as database catalog

The values are determined via an SQL query.

§ Show column name

If the values are determined via an SQL query, the column names can be shown.

The properties 'Ask for confirmation when deleting a node' and 'Ask for confirmation when creating a sub-element' are only relevant for editing catalog data. You will receive a corresponding notification.

Catalog entries

Depending on the selected properties, you can create a list catalog with values on one level or a tree or hierarchy catalog with nested values.

One tab is displayed for each language of the object definition in which you can enter a caption and tooltip text for each value. The return value is the same for values independent from the object definition language.

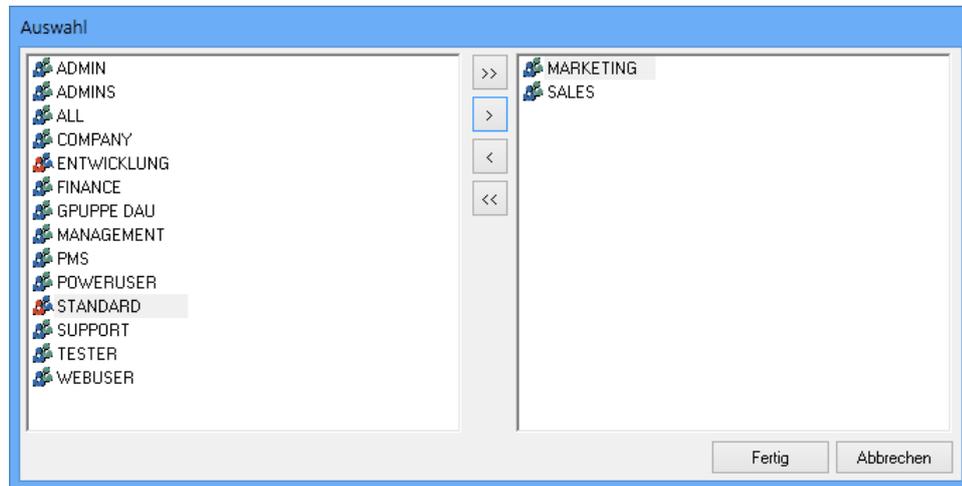
Each value can be given the 'Bold' property. It will be highlighted in the catalog with a bold font. Each value can be assigned a color.

The 'selectable' property allows the user to select and transfer a value into the field. Not selectable values are flagged with a yellow dot as a status icon or a check box with a red x. If the user can transfer several nested values, non-selectable elements are transferred as well.

The 'Expand value' option specifies that nodes should be shown expanded. This property applies to the current node that is being edited.

Use the following buttons to create and edit catalog elements:

- | | |
|---|--|
|  | <p>New element Click the button to create a new element. Enter its caption and optionally text for a tooltip. If you enter no return values, the caption is used as the return value.</p> <p>If you enter data on the tab for the default language, it will be displayed for all other object definition languages for which no alternate entries have been made.</p> <p>If elements have already been created, you can specify if you want to create a new element on the same level as the selected element (sub-element/No') or if you want to create the new element below the selected element (sub-element/Yes').</p> |
|---|--|



This is where you assign the groups whose members should be shown the values from the list of all the groups.

Field value dependency

The display of the values can be made such that it depends on the values of other index data fields.

If you have activated the setting 'Nodes depend on the value of a field,' select an index data field using the list 'Field for value dependency' when creating or editing a value and enter the value in the entry field on which the display of the catalog values should depend.

The list 'Field for value dependency' contains the index data fields of the folder type in addition to the index data fields of the object type for document types and register types.

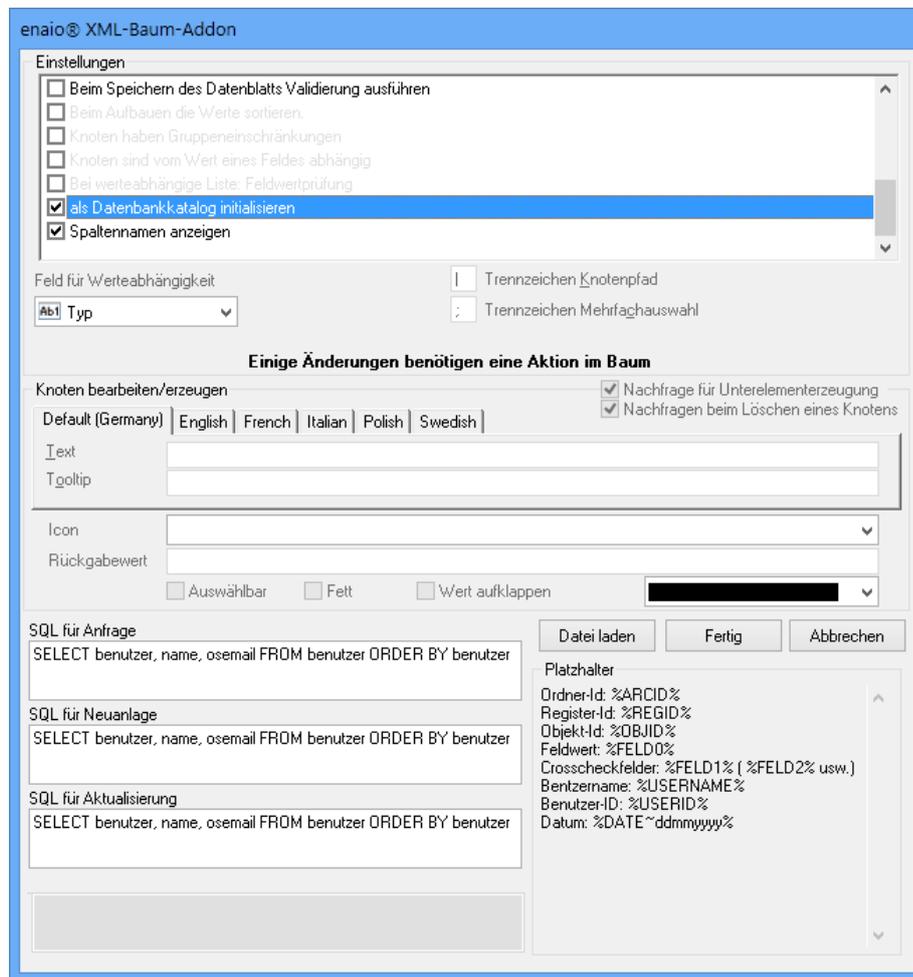
If the field for the value dependency is empty, then the value is also shown. If you activate the property 'For value-dependent list: Field value validation,' no value is shown if the field for the value dependency is empty.

SQL Queries

For catalogs with the property 'Initialize as database catalog,' the values are created via SQL queries.

You can enter different SQL queries for queries, creations and updates.

If you query several columns, the data from the first column are transferred to the AddOn field catalog. The data from the other columns can be transferred to fields with crosscheck properties.



Example:

The following SQL query produces a list of users with names, complete names, and e-mail addresses.

```
SELECT benutzer, name, osemail FROM benutzer ORDER BY benutzer
```

	benutzer	name	osemail
<input type="radio"/>	ARNAUD	Arnaud Voirot	arnaud@optimal-systems.de
<input type="radio"/>	CÉCILE	Cécile Hagmaier	cecile@optimal-systems.de
<input type="radio"/>	CLAIRE	Claire Martin	claire@optimal-systems.de
<input type="radio"/>	CLAUDIA	Claudia Schleret	schleret@optimal-systems.de
<input type="radio"/>	CLAXTON	Earl Claxton	Earl.Claxton@optimal-systems.de
<input type="radio"/>	DANNY	Danny Pawlowski	danny@optimal-systems.de
<input type="radio"/>	DAVID	David Bohla	bohla@optimal-systems.de

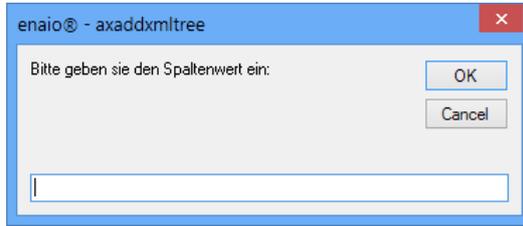
The column name display can be switched on using the corresponding property. Multiple selection is not possible.

Values with columns

Additional columns with values can be created for values. These column values can be transferred to crosscheck fields or used as comments.

How to create column values:

- § Select a value.
- § Click on the button **New element** while holding down the Ctrl key.
- § Enter the column value in the dialog.



§ Confirm with **OK**.

The column values are assigned the value.



The column values can be edited and deleted using the corresponding buttons.

When you open the AddOn catalog, the values are shown as follows:



The value can be transferred by the user into the AddOn field catalog, which transmits column values to crosscheck fields.

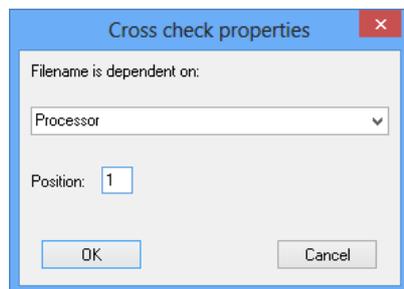
Column values apply for all languages.

Catalog AddOn and crosscheck fields

Values from columns and values from multiple-column SQL hit lists can be transferred into other fields in the data sheet. To do this, the fields need the property 'Controlled by cross check'.

When configuring the crosscheck property, specify the AddOn field catalog and the position.

For column values, the first column values are transferred via position 1, for SQL hit lists, the second column values are transferred via position 1.



COM interface

The following example script defines a catalog with two entries which are displayed for the user. The entry 'A' is preset.

```

Set ObjList = CreateObject("axaddxmltree.AsAddon")
If ObjList Is Nothing Then
    MsgBox "The basic library "axaddxmltree.dll" could not be
    initialized." + vbCrLf + "Check your installation.", vbInformation +
    vbOKOnly + vbSystemModal, "XFR"
Else
    '=====
    'The AddOn catalog is executed and evaluated
    '=====
    Dim xmlstring as String
    xmlstring = ""
    xmlstring = "<axaddxmltree returnsinglevalue='1' multiselect='1'
withcheck='1' initalist='1' initaldialog='0' selectlastnode='0'
deselectwhencollapse='0' group_splitchar=';' gridlines='0'>"
    xmlstring = xmlstring & "<ebene spezialicon='PERSON' selectable='1'
returnvalue='A' bold='0' forecolor='0' name='A' tooltip='A'/>"
    xmlstring = xmlstring & "<ebene spezialicon='PERSON' selectable='1'
returnvalue='B' bold='1' forecolor='0' name='B' tooltip='B'/>"
    xmlstring = xmlstring & "</axaddxmltree>"
    sValue = "A"
    ObjList.InitFromString sValue, xmlstring
    Set ObjList = Nothing
End If
    
```

Database AddOn

The database AddOn queries a database table of an external database and transfers the data into the index data field and connected fields.

Enter the following in the configuration field via additional entries:

- § Connection parameters to the database
- § SQL statement for data query
- § Table columns of the hit list which are not displayed
- § Table column according to which the hit list content is sorted
- § Column name for the header
- § Table columns for which their content is not transferred

Select the file `axaddfdb.dll` via the file selection dialog. It is found in the OS directory `\client32`.

You number the extra entries on the **AddOn** tab consecutively, beginning with **EXTRA00**. The entry order is irrelevant.

Example:

<p>EXTRA00=DBCCONNECT=Provider=sqloledb; Data Source=testdb,1433; Initial Catalog=testcatalog; User Id=testUser; Password=testPassword</p>	<p>Enter the connection data for the database including user and password. Security notice: these data are saved unencrypted in the object definition file. Prevent this file from unauthorized access.</p>
<p>EXTRA01=SQL=select vorname, name, benutzer, id from benutzer</p>	<p>Use a Select-statement to specify which data is retrieved from which table.</p>

EXTRA02=HIDECOLUMNS=3	You can also specify columns which will not be displayed. The first column is numbered with 0. The contents of columns which are not displayed can be transferred.
EXTRA03=SORTCOLUMN=2	You can also specify the column according to which the hit list is sorted. The first column is numbered with 0.
EXTRA=04=SORTDESCENDING=1	Sort descending; sort ascending if without entry or with a value of '0'.
EXTRA04=COLTITLE0=7,Vorname;9,First Name;12,Prenom EXTRA05=COLTITLE1=7,Name;9,Name;12,Nom EXTRA06=COLTITLE2=7,Benutzername;9,Username;12,Nom de utilisateur	Optional column names for hit list. A name can be specified for each language using a primary language identifier.
EXTRA07=IGNORECOLUMNS=0	You can also specify the columns which will not be transferred. The first column is numbered with 0. More than one entry is separated by a comma.

The following primary language identifiers are used as column names:

Language	Primary Language Identifier
German	7
English	9
Spanish	10
French	12
Hungarian	14
Italian	16
Dutch	19
Polish	21
Swedish	29

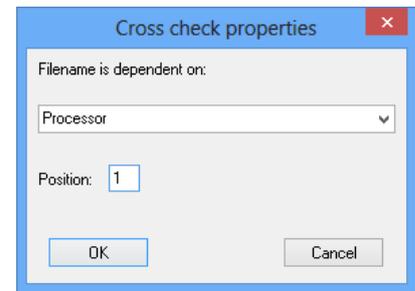
Users which have selected a language for the object definition will see the column names corresponding to each language.

The content of the first column of the row which the user has selected by double-clicking is transferred into the AddOn field.

The content of the following columns can be transferred into fields with the property 'Controlled by cross check'.

Example:

The field 'First name' depends on the AddOn field 'Name' and receives position 1. Thus the value of the first column which follows the column assigned to the AddOn field will be applied to the field 'First name'.



All other fields to which you want to apply the data have to receive the property 'Controlled by Cross Check' and consecutive numbers starting from 1 depending on the AddOn field.

VB Script AddOn

The VB script AddOn allows execution of VB scripts, for example, to edit the indexing of fields.

VB scripts can be delivered by OPTIMAL SYSTEMS GmbH on request or can be created with the VB editor. The AddOn must be licensed with the module 'VBX' and the VB editor requires the module 'VBE'. For the VB editor, the user needs the right to 'Start enaio® editor' which is set in enaio® administrator.

If you select the **AddOn** item in the Catalog line of the properties window, the **AddOn** tab will be created.

Select the file `axaddvbs.dll` via the file selection dialog. It is found in the OS directory `\client32`.

In the configuration field you enter the path to the script:

```
EXTRA00=e:\os\vbscripts\scriptname.vbs
```

If the VB Script AddOn is assigned to an index data field, the field receives a catalog button. By pressing the left mouse button, the connected script is executed, whereas the right mouse button will start the VB editor.

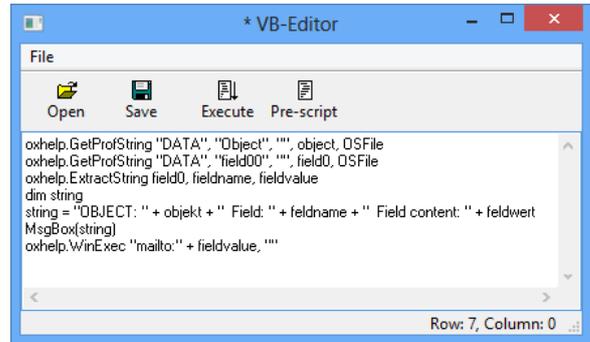
If the AddOn is placed on a form which is opened as read-only, it is only executed if you have specified `CANLOCK=0` as the last entry in the configuration field.

The VB Editor

In VB Editor scripts can be created or available scripts can be imported. Scripts which have not been provided by OPTIMAL SYSTEMS GmbH are to be created or imported and saved in VB Editor. They are encrypted while saving and can only be executed encrypted by the AddOn. Scripts can be saved with a password and require additional licensing.

The VB editor can be opened by right-clicking the set up catalog button.

In the VB editor, the library `oxactive.dll` provides the variable 'oxhelp' and the constant 'OSFile,' which enable access to index data fields. This expands the possibilities for script control enormously.



The constant 'OSFile' enables access to the handoff file.

The file has the following structure:

[DATA]

Object= Name of the object (cabinet') connected to the AddOn.

Flags= various flags, which can be set with the Editor.

Index= a DB index

Field00 till FieldXX= Names and field contents of the connected dialog fields

EXTRA00 till EXTRAXX= Extra entries from AddOn definitions.

The variable 'oxhelp' makes the following functions available:

```
GetProfString(section, key, default return string, return string, file name')
```

This function corresponds to the 'GetPrivateProfileString' Windows API function.

Example:

```
dim b, a
```

```
set b = CreateObject("oxactive.COxHelp.1")
```

```
b.GetProfString "Section", "Key", "", a, "d:\temp\test.cfg"
```

```
WriteProfString(section, key, value, file name)
```

This function corresponds to the 'WritePrivateProfileString' Windows API function.

```
WinExec(File, Parameter)
```

With this feature you can execute a Windows application.

```
ExtractString(string, Fieldname, Fieldvalue)
```

This function is used to separate the field name and the field value from the handoff file for the AddOn.

Example:

```
ExtractString("Mainconstructiongroup|Fireprotection", a, b)
```

Here, a = 'Mainconstructiongroup' and b = 'Fireprotection'

The following sample script reads values from the handoff file, displays the values, and enters them in the addressee field of the opened e-mail dialog:

```
oxhelp.GetProfString "DATA", "Object", "", object, OSFile
oxhelp.GetProfString "DATA", "field00", "", field0, OSFile

```

```

oxhelp.ExtractString field0, fieldname, fieldvalue
dim string
string = "OBJECT: " + object + " Field: " + fieldname + " Field
content: " + fieldvalue
MsgBox(string)
oxhelp.WinExec "mailto:" + fieldvalue, ""

```

Multi-Fields

With multiple fields the user can individually index the pages of a black and white, color, or grayscale document.

Via the text fields in enaio® client, a button for indexing is assigned which opens the indexing dialog. Multi-fields on a form are always grouped in one section of the dialog which is always located on the bottom part of the form.

This section is set up via the button **Multiple field section** when you create a document type. Then the section will be inserted with the standard height. The height can be set to a maximum of 5000 points. The width of the section will always be automatically adjusted to the height.

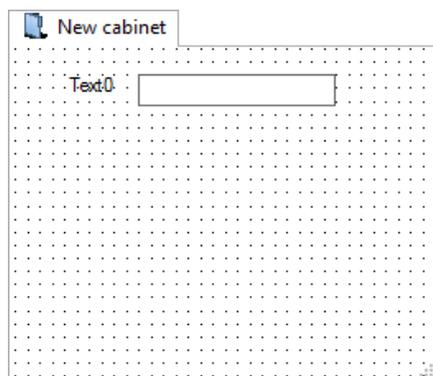
You assign the database field type 'all characters' to the multi-fields via the properties window. The further properties 'read-only', 'read-only after archiving', 'read-only after initializing' and 'Supervisor field' are available.

If the user edits the indexing of a multiple field, the corresponding database table will be locked until editing is finished. As long as the document is locked, any other client must wait until it is unlocked - no other actions will be executed. Other users will not be able to see why the client is waiting in such case. Therefore multi-fields should be used with consideration. Generally speaking, you can achieve a similar effect using the 'table' dialog element.

Layout of Forms

The dialogs are best designed with the mouse and the layout features in the work area.

Dialogs are opened in a window on the work area. In the tab label you find the object name. You can switch on the grid. Elements that you click on will have a dotted frame and handles.



If you click in the dialog you can use the mouse and the handles to adjust the width and the height. The dialog then has a ↔ horizontal, a ↓ vertical, and a ↘ diagonal sizing handle. When dragging beyond the border of the window; the window will be automatically adjusted. Via the settings, you can specify the maximum and the minimum dialog size.

All mouse actions snap to the grid.

You can show/hide the grid and edit the grid size:

 Open the dialog **Grid settings** via the button.



Enter a value for Width and Height of the grid blocks.

Select whether the grid 

The grid size of '1 x 1' turns the grid off.

Confirm the changes with **OK**.

The default grid settings are selected via the menu **File / Settings**. The settings are saved for each individual user.

Multi-fields should only be created in multiple sections and not moved from there.

Selecting and Positioning Dialog Elements

Dialog elements are created using the **Dialog elements** toolbar (see 'Creating Dialog Elements').

If you click on a created dialog element on the work area or the workspace, it will be opened with a dotted frame and handles.

Text fields consist of two graphic elements, an input field and a label field. Both can be selected and  moved independently from each other. If you hold the Shift key while selecting an element the other element will also be selected with sizing handles but not dotted.

Multiple graphic elements can be selected with the mouse by holding the Ctrl key. Each newly-selected element will appear with handles and a dotted frame. You can also hold the mouse button to drag a rectangle around all elements you want to select. If you then click an element, it will also be given a dotted frame.

The size adjustment and alignment functions refer to the position of the dotted element.

If you click on an already-selected object while holding the Ctrl key, the selection will be removed.

Hover the mouse above one of the selected objects to move them all together based on the grid , or use the cursor keys to move them point by point.



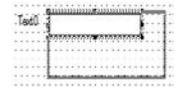
If you have selected multiple elements, the properties window will only contain those properties which all elements can have edited together.

On the **Edit** toolbar, you also find buttons for cutting, copying, and pasting dialog elements.

Size of Dialog Elements

The size of an element is changed using the sizing handles. Selected elements have ↔ horizontal, ↑↓ vertical, and ↗↘ diagonal handles.

If you click and drag an element larger or smaller, it will snap to the size of the grid.



The size of graphics cannot be changed.

If you enlarge or shrink the input field of a text field, the property 'normal/multiline' will not change.

You can adjust the label size of text fields, static text, radio buttons, and check boxes located in an element by dragging. The elements themselves are set to default size. The field size is the reference point for the fitting of multiple elements.

 The field size can be adjusted via the **Match size with content** button on the toolbar or the **Layout** toolbar.

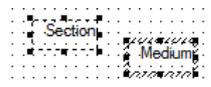
 Multiple selected elements can be adjusted in height and width using the **Match size** button, the **Match height** button, and the **Match width** button. The reference size is always the size of the element with the dotted frame.



Aligning Dialog Elements

The positions of the dialog elements are aligned with the help of the features in the **Layout** toolbar.

When aligning, all elements are aligned according to the field or element which has a dotted frame.



You can align the left, lower, the right or the lower frame of the selected element to the element with the dotted frame.

You can center the elements horizontally or vertically. The reference point is the element with the dotted frame.

With the **Layout/Arrange** menu, you can vertically and horizontally unify the distance between selected elements. You can choose to execute the function based either on the distance between the center-lines of the elements, or the distance between the elements themselves.

Please note that there is no difference made between label fields and input fields when aligning and arranging. If both are selected they will be aligned according to the same reference point.

Copying, Inserting, and Deleting Dialog Elements

Selected dialog elements can be cut or copied with the context menu or the buttons on the standard toolbar in order to place them anywhere else in the dialog or on a different dialog.

- § Select the intended elements.
 - § Choose **Cut** or **Copy** from the context menu.
 - § Open the destination dialog (if not the same).
 - § Choose **Paste** from the context menu.
- The cursor is used as a \updownarrow positioning tool.
- § Click any position in the dialog.

The cut or copied element will be inserted at the selected position. They retain their properties.

If the dialog already contains elements with the same label, the inserted element will have a number appended.

For text fields that have catalogs or AddOns assigned to them, you must adjust the parameters after they have been inserted. Radio buttons require an associated group field.

Selected dialog elements are deleted using the context menu or the **Del** key. In text fields, both elements the label field and the input field will be removed even if you have selected only one of them.

Tab Position of Dialog Elements

The user can activate the dialog elements in enaio® client with the tab key. When created, the dialog elements automatically receive an order of the tab position in the order in which they were created. In the workspace the dialog elements are always positioned in the order of the tab position. Dialog elements which are assigned a page control on a page have a tab position relative to the page.

The tab order can be edited as follows:

- § Open the dialog of an object from the workspace.
- § Switch on the tab mode via the button **Tab order**.

The tab order will be displayed in the dialog. It starts with position '0.'

- § Use the left mouse button to continuously increase the tab positions.
- With the right mouse key you can copy the position number which is then used as the basis for the incrementing counter.
- § Switch on the tab mode again via the button **Tab order**.

The dialog elements will be aligned in this tab order in the workspace.

A group field and the assigned radio buttons must directly follow on from each other in the tab order. The relative order of tab positions cannot be edited if data has been entered.

The dialog element 'Pagecontrol' has a position, on every page the order starts again with the position '0'.

Changes to the tab order have no effect on the database.

The Test Mode

In test mode, dialogs are displayed the way they will show in enaio® client.

The features of AddOns and multiple fields will not be displayed in test mode. For page controls, placeholders will show.

Dialogs which the user opens for searches in enaio® client or combines on the data sheets can be displayed larger than they are defined and then shown in test mode.

 Test mode is switched on via the button on the standard icon bar.

In test mode the dialog will show the way it was last saved in the workspace.

Languages for Object Definitions

Multiple Languages – Introduction

The labels of archive object types, dialog elements, and the tooltips can be specified in multiple languages using language versions. The length of the labels is still restricted to 30 characters for object types and 240 characters for dialog elements.

For the configuration of add-ons which refer to other dialog elements or object types you must use internal names so that they also work in other languages. When configuring AddOns that have references to other dialog elements or object types, you must use internal names so that these AddOns will also function in other languages. Internal names for folder, register, and document types are also required if users set different languages for the object definition. If e.g. a user sends an internal reference for an object, the recipient can only open this object in enaio® client if he uses the same language or if an internal name was specified for the object type name.

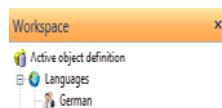
If you set up a different language, other users of enaio® client can select this language. The names for archive object types, dialog elements, and tooltips are then displayed accordingly.

Language settings have no effect on indexing.

The graphic properties of the forms and the fields are the same in all languages, but please note the possibility of different name lengths when designing the forms.

Setting up Languages

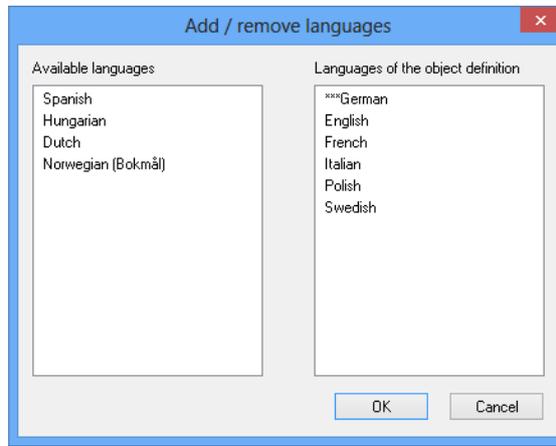
An object definition is always created in the default language 'German.'



From the workspace you can add and remove languages:

- § **Select Languages** in the workspace.
- § Select **Add / remove languages** from the context menu (menu key).

The following dialog will be opened:



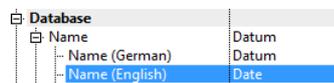
§ Select an available language and then use the arrow keys to assign the object definition to the languages.

§ Confirm with **OK**.

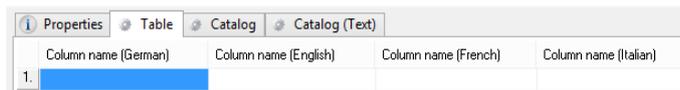
The added language will be displayed on the workspace.

§ Save the object definition.

As soon as you have added the language, all fields for labels and tooltips will be added to the tree structure as a new field in the properties dialog. There you can edit the names of the language versions.



For the dialog element 'table' a column is displayed for each added language.



Via the context menu of a form and the **Form** menu, you can select the language which should be used for the display.

If you delete the language, the language entries are permanently deleted as soon as you save the object definition.

Creation of Database Tables

Database Tables – Introduction

For every cabinet and folder and document type of the current object definition, you must create a table in the database.

The tables contain editable dialog elements for text fields and check boxes; one column for each. For text fields you enter the database type and the field length via the properties window. For checkboxes and radio buttons the properties are managed automatically.

For radio buttons which are combined by group fields, a column is created.

The database entries for the option which is selected in the client depend on the relative order of the tab positions. This relative order is not allowed to be edited after data has been entered. Otherwise you risk data loss or data may be used incorrectly.

All tables contain additional columns which are created and managed automatically, for example for base parameters.

In addition to these object tables, the database contains system tables which are created when enaio® server is installed and are then managed automatically, for example tables for the user settings. These system tables can be displayed in enaio® editor, but not be modified.

Before you update the database you should validate the object definition (see 'Validating an Object Definition')

Created or edited objects are flagged in the workspace with a '|' if they require table adaptation. When saving the object definition you will be notified about necessary table adaptations and can instantly execute them. If extensive table adaptations are necessary, the object definition should be validated first.

Table adaptation will be required in the following cases:

- § a new object type was added,
- § a new dialog element of the following type was added:
 - any text field
 - a checkbox
 - a radio button of a new group
 - a table
- § a new column was added to the 'table' dialog element,
- § the database type of a column of the 'table' dialog element was changed,
- § the database type of a dialog element was changed,
- § the database length of a dialog element was changed

If you update the database tables to match the object definitions, tables that are no longer required will not be automatically deleted but filed as unused tables.

Unused tables must be deleted before you create new tables.

Additionally you can create and delete index tables.

The results of the table check and fitting will be displayed in the output window.

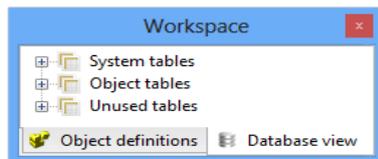
Additionally, you can enter a file in the 'Settings', to which the results should be written.

Checking Tables

In the workspace all stored database tables are divided into system tables, object tables, and unused tables and then they are displayed in a tree structure on the **Database** tab.

System tables cannot be edited. Please contact the support team if they are missing or have errors.

Object tables are displayed sorted alphabetically or  by cabinets.



Via the tree structure, you can display individual tables or columns. The tables are selected with internal database names in addition to the object names.

For every object from the active object definitions an object table is created. Every object table contains columns for the stored text fields, the check boxes, and the grouped radio buttons.

On the **Database** tab, missing tables and columns will be shown.

 Tables and columns may be missing.

If you switch on the  detailed mode, tables and columns will also be checked for errors.

Erroneous tables and columns are flagged in detailed mode, too.

 Tables and columns may contain errors.

Tables contain errors if columns are missing or there are other errors. They have errors if the properties of the active object definition are not identical to the properties of the database.

In the detailed mode, the indexed columns are selected with a red 'i'.

Displaying the tables and columns in detailed mode can require a long time for large object definitions, because every table and every column will be checked. The result of the detail check will be temporarily stored. Afterwards, only changes will be checked. If you edit single tables you can then check them individually and you do not have to check all tables and columns in the detailed mode.

One or more selected tables can be checked via the button  **Check table(s) / show details**. The result of this detailed check will be temporarily saved and when this feature is called for again the stored results will show but no new check is executed.

If you want to check the tables again, select the feature  **Check table(s) again**.

All tables will be checked when you use the detailed mode for the first time or if you use the button  **Check all tables again**.

If the tables or columns have error or are missing you have to update the tables.

The results of the check will be displayed in the output window.

If you edit the order of the tab positions between the radio buttons you will receive a notice that the column for the radio button is missing. If you create a new column the old one will be deleted, even if it contains data.

Adjusting Tables

When updating the database tables, missing object tables will be created, missing columns in the object tables will be added and object tables that have no object which applies to the object definition will be marked as unused.

Columns which are too small can be adjusted. Columns which are too large because the field length in the database is longer than in the object definition cannot be edited, they have to be newly created. If the database does not already contain tables which have field lengths that are too long, then the entire table will be deleted when it is updated and a new one is automatically created. If there is data in the table you receive a security alert when updating and you can choose whether you want to create the table or leave it the way it is. If fields are deleted, the corresponding columns will not be deleted.

You can  adjust single tables or the entire  database.

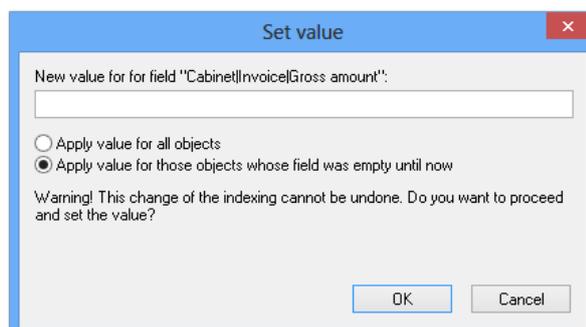
These features are accessed via the buttons on the **Database** toolbar or via the context menu on the **Database** tab. Single tables can also be adjusted via the context menu on the **Object definition** tab.

Via the feature **Select error tables**, you can mark all tables that were selected as containing errors with a check.

The results of the update will be displayed in the output window.

If you edit the order of the tab positions between the radio buttons you will receive a notice that the column for the radio button is missing. If you create a new column the old one will be deleted, even if it contains data.

If you select the column of a table in the database view, you will find the feature **Set value** in the context menu. Here you can specify a value which will be entered into the column of all fields of existing objects of this type or to all the empty fields. With this feature you can enter data from an indexing dialog into a field for an object type.



Please note that no checking of inconsistencies with catalogs, AddOns, or scripts is performed.

Removing Tables

Objects that you have deleted from the object definition, without deleting the corresponding database tables of the object type with the data, will be marked as unused tables under the **Database** tab on the workspace. These unused tables must be deleted before you create new tables.

§ Select an  unused table.

§ Select **Remove table** from the context menu or via the button on the **Database** toolbar.

The database table with the data will be deleted.

Database indexing

A database index can enormously accelerate queries of large databases.

With database indexing using enaio® editor you can index by standard or self defined criteria for table column indexing. As soon as a table column is indexed, the database continues the indexing whenever data is added.

When indexing the database, new index tables are created and existing index tables are deleted and recreated. You can also refresh the index tables after extensive editing of the archive or of the object definition file to bring them up to date.

You can only index the database if no user is accessing the database. Empty tables will not be indexed.

Using enaio® administrator you can optimize index statistics without having to create new ones. For further information refer to the 'enaio® administration' manual.

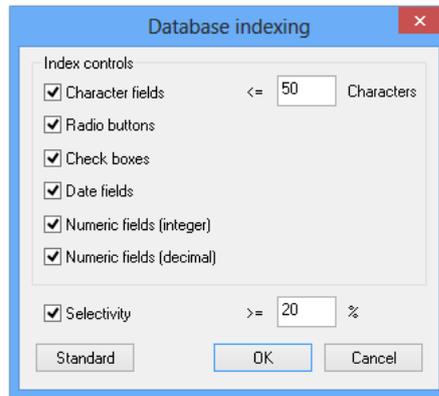
When you index with standard settings, enaio® editor creates index tables for the following fields:

- § Character fields with a maximum of 50 characters,
- § Radio buttons,
- § Check boxes,
- § Date fields,
- § Number fields.

These criteria for index tables do not always have a positive effect on performance. Therefore, you can create index tables based on your own criteria.

§ Select the entry **Indexing/settings** from the **Database** menu.

The window Database indexing will be opened.



The standard criteria are set by default. These can be recovered via the **Restore standard** button.

- § Select the fields for which an index table should be created.
- § Confirm with **OK**.

The settings are saved.

- § Select the entry **Indexing/start** from the **Database** menu or click the button  **Database indexing**.

The index tables will be created or deleted according to your criteria.

For character fields you additionally specify the number of characters. The columns of text fields which are larger than specified will not receive a table index.

The selectivity specifies the relationship between the number of different entries in a field and the number of documents.

It is calculated as follows:

Number of different entries x 100 / Number of documents

Example: 300 different entries x 100 / 500 Documents = 60%

The higher the selectivity is, the more sense it makes to create index tables. For long character fields which usually have high selectivity, the duration of time to enter the running indexing can exceed the advantage of using the index tables for the queries. The criterion of the selectivity only makes sense if you have a representative number of documents by which you can calculate the selectivity.

Field selection and selectivity are combined by a logical AND.

Recommendation: The indexing of character fields which are important to the query – for example customer name, customer number etc. – usually makes sense. The indexing of check boxes and radio buttons usually does not make sense.

Indexing can cost a lot of time. In the detailed view, the indexed columns are selected with an 'i'.

Instead of the indexing with enaio® editor you can save an SQL script with the indexing settings and load this script using the database tools, optimizing indexing. An SQL script can be created via the menu **Database / indexing**.

Instead of indexing all tables you can select a table and index it via the context menu. In this case, the current settings of the database indexing dialog are used.

In the database view you can mark single columns and then use the context menu to edit the indexing properties and create and delete indexes.

Upon indexing from enaio® editor, all indexes which correspond to the naming scheme for the respective object table type are deleted.

For folder tables: 'Sn_' with 'n' being the number of the folder table. Example: The naming scheme for indexes for object table 'stamm11' is 'S11_'.

For register tables: 'Rn_' with 'n' being the number of the register table. Example: The naming scheme for indexes for object table 'register2' is 'R2_'.

For document tables: 'On_' with 'n' being the number of the document table. Example: The naming scheme for indexes for object table 'object22' is 'O22_'.

If you create indexes for the respective table, independently from enaio® editor, the name of the index cannot correspond to the naming scheme of the object table, in order to prevent the deletion of these indexes in enaio® editor.

Index

A

AddOn
 read-only 62
 verifying 62
 add-ons 60
 annotations on layers 19
 Application AddOn 74
 archiving document versions 19

B

buttons 29, 48

C

cabinet 14
 catalog AddOn 79
 catalogs 48
 check boxes 28
 checking tables 100
 COM functions 16
 constants 37
 conversion AddOn 66
 creating dialog elements 29
 customizing tables 101

D

database catalog 55
 database field length 27
 database field type 32
 database index 25
 database indexing 102
 date 32
 date AddOn 65
 default
 constant 37
 functions 37
 defaults 9, 37
 defining a form 35
 Dialog element types 27
 dialog elements 27
 document history 19
 document type without pages 18
 document type without slides 18

E

editing history 19, 30
 e-mail document types 21
 Embedded objects 21
 encryption 19
 enforce security system 16

F

filing plan add-on 66
 folder 14
 form 34
 full text indexing 16

G

grid settings 92
 group field 28
 group fields 42

H

hierarchy catalog 54

I

icon catalog 51
 icons 21
 ID AddOn 71
 image 29, 47
 importing an object definition 13
 index data history 16, 19
 installation 6
 internal names 15, 31

K

key field 36

L

licensing 6
 linked to W-document 37
 list catalog 49
 list icons 51
 logging 10

M

matchmode 54
 modules 17
 module-spanning document type 18
 multiple field 91

O

object definition file 11
 object definitions 11
 object relations 23
 object tables 99
 object type relations 23
 optimizing index statistics 102
 output window 8

P

page control 29
 performance wizard 25

Q

query AddOn 74
Quickfinder AddOn 62

R

radio buttons 28
reference type 16
Removing tables 102
required field 35

S

search group 38
security system 6, 16
selectivity 103
separator 54
server engines 12
setting values 101
settings
 default grid settings 9
 maximum database field length 9
 min / max dialog size 9
 miscellaneous 9
SQL script 103
static text 29, 47
structure tree 56
system role 6
system tables 99

T

tab position 94
table 29, 43
test mode 95
text field 28
toolbar 7
tooltips 27
tree catalog 53

U

unused tables 99
user AddOn 71, 73
user interface 6

V

validating an object definition 24
VB Editor 89
VB script AddOn 89

W

workspace 7
WWW AddOn 65