



White Paper

Fabasoft Folio Forms

Fabasoft Folio 2022

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2022.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Contents

1 Introduction	4
2 Software Requirements	4
3 Disambiguation	4
4 Selection	4
5 Determination of Form Pages	5
5.1 Display Inherited Pages = No/Undefined	5
5.2 Display Inherited Pages = Yes	5
5.3 Generic View for Derived Classes = No/Undefined	5
5.4 Generic View for Derived Classes = Yes	5
5.5 Generic View = No	5
5.6 Generic View = Yes	5
6 Properties on Generic Pages	6
7 Examples	6
7.1 Example 1	6
7.2 Example 2	8
7.3 Example 3	8
7.4 Example 4	10
8 Limitations for Forms that can be Passed via URL	11

1 Introduction

Forms (`COOATTREDIT@1.1:ObjectEditForm` and `COOATTREDIT@1.1:ObjectDeskForm`) are used to divide properties on multiple tabs (dataset) and to align properties on these tabs (layout).

Which form is used is specified in the object class, in which different criteria are used, for example the applicable action (e.g. to distinguish forms for reading and forms for editing), the seeking authorization (to set the user group for the form) and other conditions, with which the usability of a form can be effected.

In compiling the form pages (tabs) it is possible to use the object class hierarchy. On the one hand in a form can be defined to take the form of the basic classes into account (*Display Inherited Pages* - `COOATTREDIT@1.1:forminheritance`). On the other hand it is possible that the form system is creating automatically generated form pages for properties of an object class (*Generic View* - `COOATTREDIT@1.1:formgeneric`).

The behavior of how the automatically generated pages (generic pages) are created has changed in Fabasoft Folio 2010 Spring Release compared to previous releases. There is a new property *Generic View for Derived Classes* (`COOATTREDIT@1.1:formgenericderived`).

Before Fabasoft Folio 2010 Spring Release, generic pages were created when no form was found in an object class. From Fabasoft Folio 2010 Spring Release a form is determined on the basis of the object class hierarchy in all cases. This form will then determine which pages are being added generically.

In this document is explained how generic pages are created.

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

3 Disambiguation

- Initial class: Object class from which the search for the form is begun.
- Form class: Object class in which a form is found.
- Form: Form that is found in a form class.

4 Selection

If no form is found in the initial class, the object class hierarchy is searched to top, until a class with a form is found (form class).

As before, in Fabasoft Folio the known criteria are applied in the selection of forms: search permission, action, conditions of the property *Used if* (`COOSYSTEM@1.1:viewcondition`).

If no form is found, nothing is displayed.

Note: Also the initial class can be the form class.

5 Determination of Form Pages

In the form found in the form class, the pages are determined depending on the following properties:

- *Display Inherited Pages* (COOATTREDIT@1.1:forminheritance)
- *Generic View* (COOATTREDIT@1.1:formgeneric)
- *Generic View for Derived Classes* (COOATTREDIT@1.1:formgenericderived)

5.1 Display Inherited Pages = No/Undefined

The base classes of the form class are not searched for further forms. The settings of the base class are not relevant for this form.

5.2 Display Inherited Pages = Yes

In addition to the pages defined in the form of the form class, the base classes are searched for further forms. The direct base class of the form class is regarded as new initial class.

Properties directly assigned to the form class but not used on a certain page of the form are not displayed. So for the form class **no** generic page is created.

5.3 Generic View for Derived Classes = No/Undefined

Properties assigned to an involved class (from the initial class to the form class, which is not included) that are not used on a certain page of the form are not displayed. Thus, for the initial class **no** generic page is created.

5.4 Generic View for Derived Classes = Yes

The pages of the form class form are displayed. Additionally **one** generic page is displayed which contains all properties assigned to the involved classes (from the initial class to the form class, which is not included). This page gets the name of the initial class and is displayed before the other pages of the form.

5.5 Generic View = No

Properties that are assigned to the form class, but are not used on a certain page of the form, are not displayed. Thus, for the form class **no** generic page is created.

5.6 Generic View = Yes

The pages of the form class form are displayed. Additionally **one** generic page is displayed which contains all properties assigned to the form class. This page gets the name of the form class. If the form class form contains at least one page, the term "Advanced" is added to the name of the generic page. In the form class, in the property *Multilingual Name for Editor* (COOSYSTEM@1.1:classifieditmlname), another name can be defined for the generic page. This generic page is displayed after the other pages of the form.

6 Properties on Generic Pages

On generic pages only properties are displayed, that are not found on the pages located in front in the dataset. It follows:

- A property is displayed on a generic page, even if the property is listed in a form of a base class.
- If a property is hidden due to a visible-expression, it is not displayed on the generic page.
- Page filter are considered in the calculation of generic forms (see property *Display Items Filter* (COOATTREDIT@1.1:formpageitemaction) in *Form Pages* (COOATTREDIT@1.1:FormPage)).

7 Examples

Given the following object class hierarchy:

- Object class *Object* with the following properties:
 - *Name*
 - *Created on/at*
 - *ACL Object*
 - *Version Number*
 - *Recorded*
 - *Recorded Set on/at*
- Object class *Component Object* derived from *Object* with the following properties:
 - *Reference*
 - *Software Component*
 - *Private*
 - *Obsolete*
 - *Deleted*
- Object class *Configuration Object* derived from *Component Object* without properties
- Object class *Position* derived from *Configuration Object* with the following properties:
 - *Multilingual Name*
 - *Template Collections*
 - *Activities*
 - *Signature Regulations*

7.1 Example 1

Form settings:

- Generic forms on all classes with properties.
- In the object class *Object* the page name "Additional Properties" is defined for the generic page.

Class	Form	Form Pages	Properties on the Page
<i>Position</i>	FormPosition generic = yes	Position	<i>Multilingual Name</i>

	inherit = yes genericderived = irrelevant		
<i>Configuration Object</i>			
<i>Component Object</i>	FormCompObj generic = yes inherit = yes genericderived = yes	Component Object	<i>Reference</i> <i>Software Component</i>
<i>Object</i>	FormObject generic = yes inherit = irrelevant genericderived = yes	Object Security Versions	"Object" page: <i>Name</i> <i>Created on/at</i> "Security" page: <i>ACL Object</i> "Versions" page: <i>Version Number</i>

Page	Properties	Notes
Position	<i>Multilingual Name</i>	
Position (Advanced)	<i>Template Collections</i> <i>Activities</i> <i>Signature Regulations</i>	Generic page with remaining properties of the object class <i>Position</i>
Component Object	<i>Reference</i> <i>Software Component</i>	
Component Object (Advanced)	<i>Private</i> <i>Obsolete</i> <i>Deleted</i>	Generic page with remaining properties of the object class <i>Component Object</i>
Object	<i>Name</i> <i>Created on/at</i>	
Security	<i>ACL Object</i>	
Versions	<i>Version Number</i>	

Additional Properties	<i>Recorded</i> <i>Recorded Set on/at</i>	Generic page with remaining properties of the object class <i>Object</i> Page name is defined in the object class
-----------------------	--	--

7.2 Example 2

Form settings:

- No forms in initial classes.
- No generic pages.
- Only one default form in the base class *Object* (e.g. for end users).

Class	Form	Form Pages	Properties on the Page
<i>Position</i>	-		
<i>Component Object Without Reference</i>	-		
<i>Configuration Object</i>	-		
<i>Component Object</i>	-		
<i>Object</i>	FormObjectDefault generic = no inherit = irrelevant genericderived = no	Object	<i>Name</i> <i>Created on/at</i>

Page	Properties
Object	<i>Name</i> <i>Created on/at</i>

7.3 Example 3

Form settings:

- No forms in initial classes.
- Only one default form in the base class *Object* and generic pages (e.g. for administrators).
- In the object class *Object* the page name "Additional Properties" is defined for the generic page.

Class	Form	Form Pages	Properties on the Page
-------	------	------------	------------------------

<i>Position</i>	-		
<i>Component Object Without Reference</i>	-		
<i>Configuration Object</i>	-		
<i>Component Object</i>	-		
<i>Object</i>	FormObject generic = yes inherit = irrelevant genericderived = yes	Object Security Versions	"Object" page: <i>Name</i> <i>Created on/at</i> "Security" page: <i>ACL Object</i> "Versions" page: <i>Version Number</i>

Page	Properties	Notes
Position	<i>Multilingual Name</i> <i>Template Collections</i> <i>Activities</i> <i>Signature Regulations</i> <i>Reference</i> <i>Software Component</i> <i>Private</i> <i>Obsolete</i> <i>Deleted</i>	Generic Page for all properties of classes <i>Position</i> to <i>Component Object</i>
Object	<i>Name</i> <i>Created on/at</i>	
Security	<i>ACL Object</i>	
Versions	<i>Version Number</i>	
Additional Properties	<i>Recorded</i> <i>Recorded Set on/at</i>	Generic page for remaining properties of the object class <i>Object</i> Page name is defined in the object class

7.4 Example 4

Form settings:

- No forms in initial classes.
- In object class *Component Object*, `FormGeneric` is used as form (thus generic page without additional pages).
- In the object class *Object* the page name "Additional Properties" is defined for the generic page.

Class	Form	Form Pages	Properties on the Page
<i>Position</i>	-		
<i>Component Object Without Reference</i>	-		
<i>Configuration Object</i>	-		
<i>Component Object</i>	FormGeneric generic = yes inherit = yes genericderived = yes	No pages	
<i>Object</i>	FormObject generic = yes inherit = yes	Object Security Versions	"Object" page: <i>Name</i> <i>Created on/at</i> "Security" page: <i>ACL Object</i> "Versions" page: <i>Version Number</i>

Page	Properties	Notes
Position	<i>Multilingual Name</i> <i>Template Collections</i> <i>Activities</i> <i>Signature Regulations</i>	Generic page for all properties of object classes <i>Position</i> to <i>Configuration Object</i>
Component Object	<i>Reference</i> <i>Software Component</i> <i>Private</i>	Generic page for properties of object class <i>Component Object</i>

	<i>Obsolete</i> <i>Deleted</i>	Page name without "Advanced" because the form does not contain pages
Object	<i>Name</i> <i>Created on/at</i>	
Security	<i>ACL Object</i>	
Versions	<i>Version Number</i>	
Additional Properties	<i>Recorded</i> <i>Recorded Set on/at</i>	Generic page for remaining properties of the object class <i>Object</i> Page name is defined in the object class

8 Limitations for Forms that can be Passed via URL

If a form is passed as a parameter when displaying the attribute editor, by calling the action `FSCVENV@1.1001:CheckFormSecurity` is checked whether the current user has the necessary access rights for this form.

This action checks the following conditions which have to be met to display the form:

- The current user must have the search permission for the form which is to be displayed.
- The form must not be generic.
- The form must not contain inherited pages.

If one of these conditions is not fulfilled, an error message is displayed, that the form must not be used.