

Customizing ENOVIA V6 : Wörterbuch anlegen

Teil 1 : Erstellen eines neuen Typen und Einbindung in den 'Type Chooser' der Standardsuche

Inhalt in diesem Kapitel

- Neue Attribute im BUSINESS anlegen
- Neuen Typ im BUSINESS anlegen
- Erstellen und Zuweisen einer neuen Policy
- Erzeugen eines neuen Wörterbuch Eintrages im MATRIX
- Anpassen des 'Type Chooser' für die webbasierende Suche

Erklärung :

Der folgende Text stellt lediglich persönliche Erfahrungen dar und ist keine offizielle Dokumentation.

Alle hier erwähnten Produkt- und Firmennamen sind Marken der jeweiligen Eigentümer.

Einführung

Zu Beginn verwenden Sie das BUSINESS um neue Attribute, Type und Policy zu erzeugen. Wie dies gemacht wird, ist kein Geheimnis und wurde von Ihnen bestimmt schon das ein oder andere mal durchgeführt.

Für das Dictionary benötigen Sie :

drei Attribute:

Description_E (type String / Multiline)

Description_F (type String / Multiline)

Description_G (type String / Multiline)

Einen neuen Type:

Dictionary (not derived from any other / inherited the three previous created attributes)

Eine neue Policy:

Dictionary (governed the type Dictionary / one State 'Exists' / allstates and full access)

Um eine Verbindung zwischen einer Policy und einem neuen Type zu definieren, editieren Sie die Policy und wählen auf der Seite 'Governed Types' den gewünschten Type aus.

Achten Sie darauf, dass Sie auch die entsprechenden Symbolic Names erstellen.

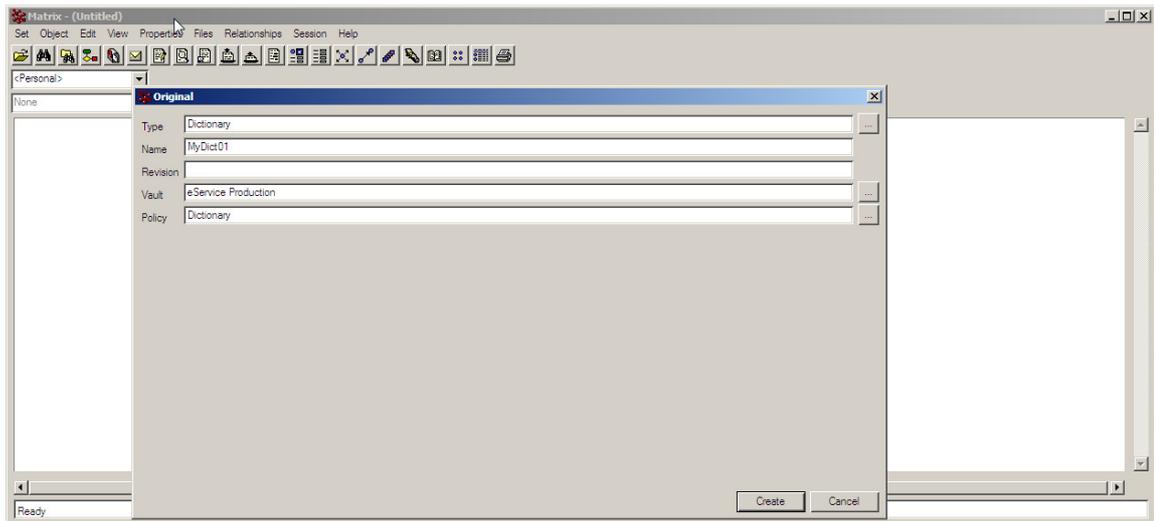
Im Anhang dieses Posts finden Sie die MQL Kommandos zur Erzeugung der Attribute, Type und Policy.

Customizing ENOVIA V6 : Wörterbuch anlegen

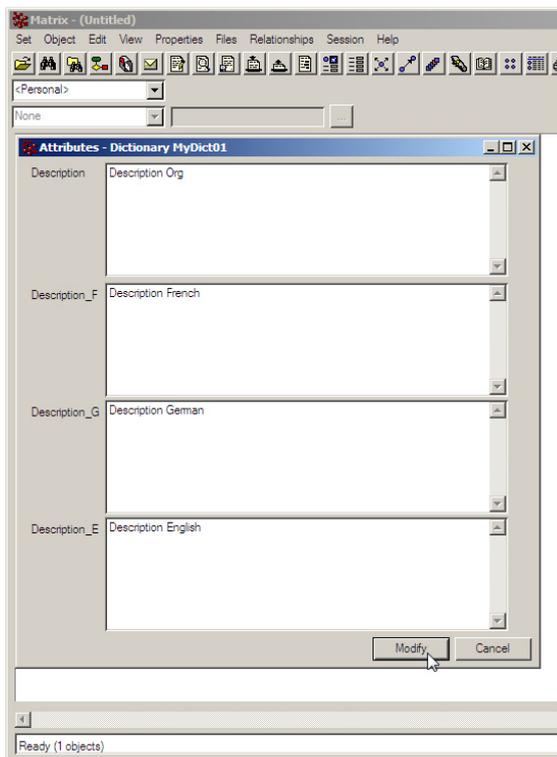
Nachdem Sie das Datenbankmodell im BUSINESS geändert haben, kann das Wörterbuch direkt in MATRIX mit Einträgen gefüllt werden. Ebenso ist das Suchen nach einem Wörterbucheintrag in MATRIX direkt möglich

Einen neuen Eintrag im MATRIX rich client erstellen

1. Erstellen Sie ein neues Object / Original und wählen den Typ 'Dictionary' aus der Type-Liste aus. Legen Sie einen Namen fest und wählen die Policy 'Dictionary' aus. [Create]



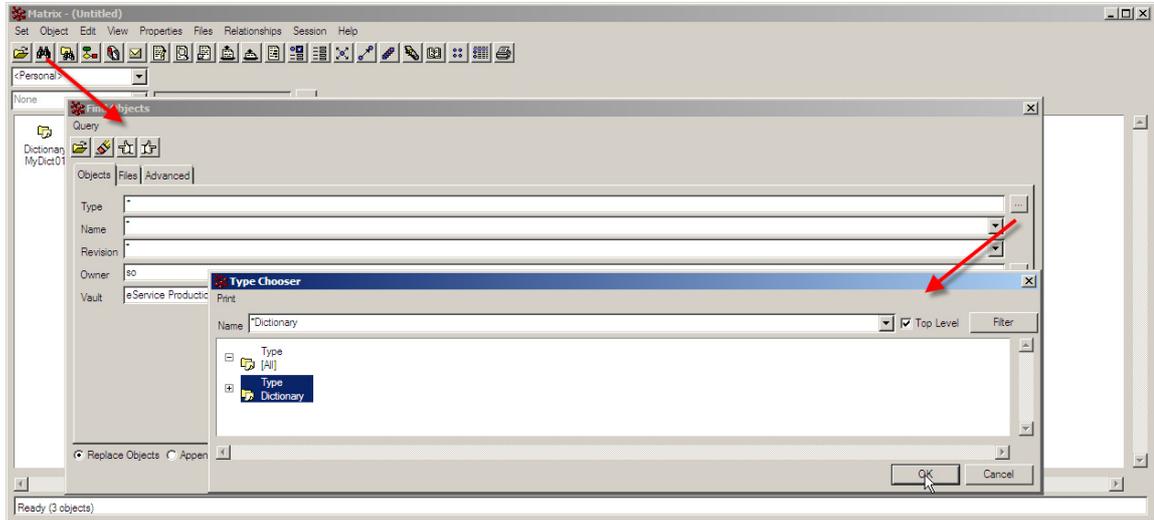
2. Füllen Sie auch die zusätzlichen Attribute aus [Modify]



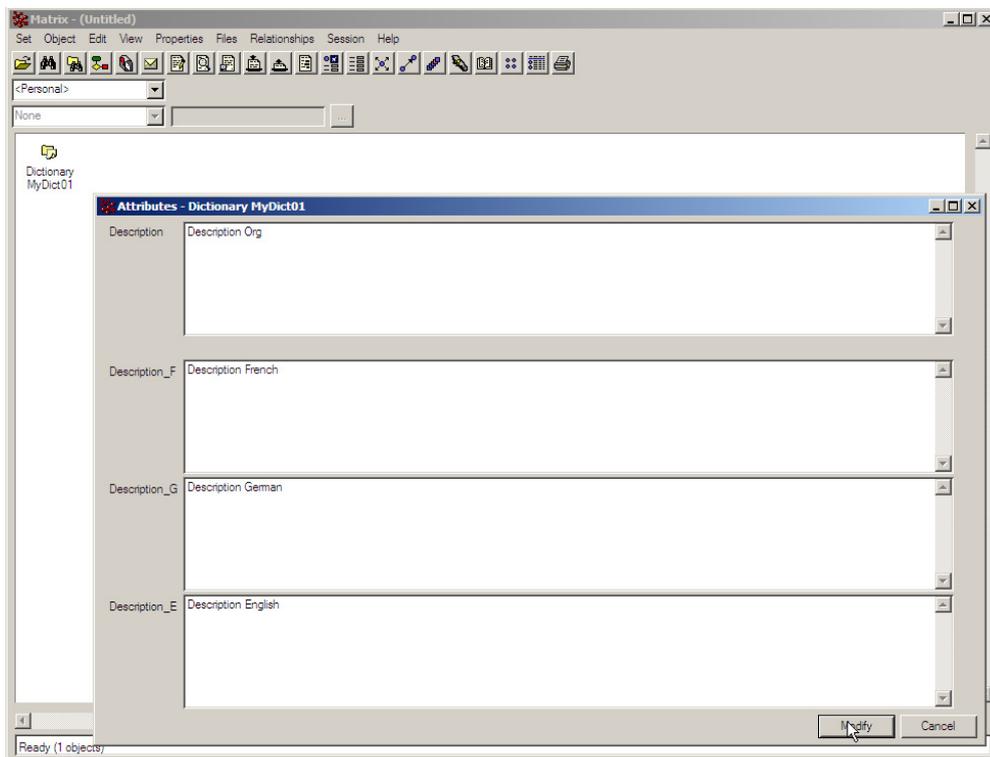
Customizing ENOVIA V6 : Wörterbuch anlegen

--> der neue Wörterbucheintrag wird in der Datenbank gespeichert

3. Sehen Sie, dass in der Suche der neue Type 'Dictionary' ebenfalls in der Type-Auswahlliste des MATRIX rich client aufgelistet wird.



--> Führen Sie die Suche aus und schauen sich den zuvor erstellten Wörterbucheintrag an

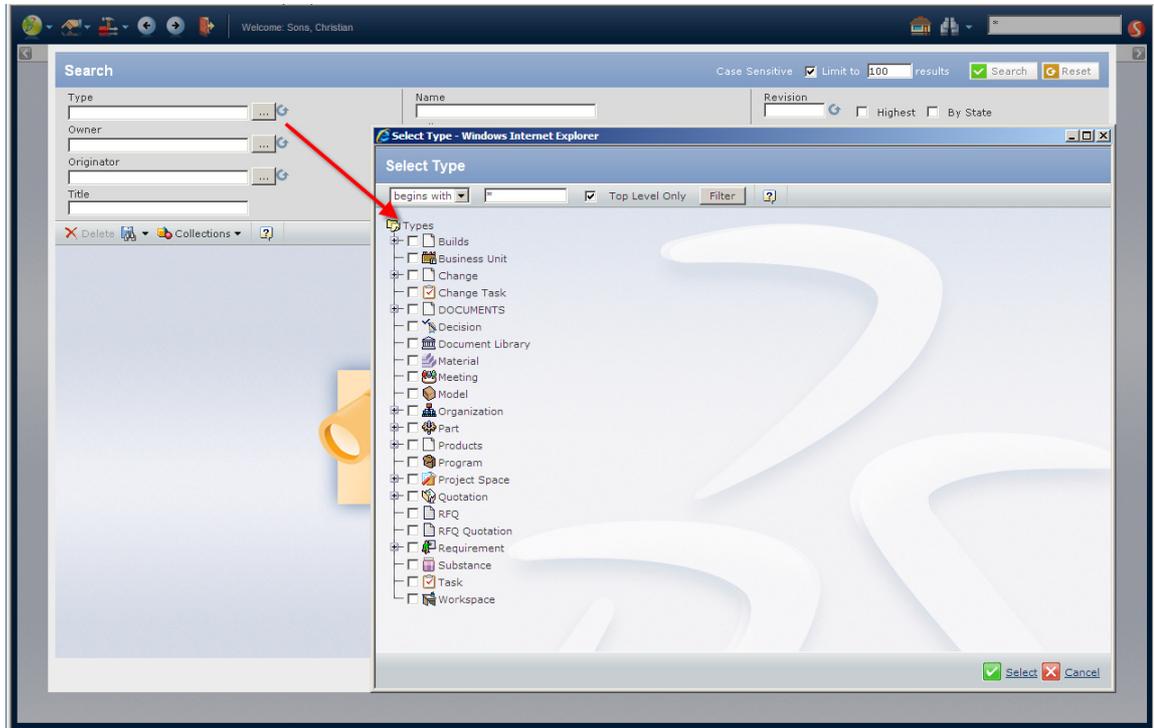


Type Auswahl im Web Interface

Customizing ENOVIA V6 : Wörterbuch anlegen

Aber was ist mit der Auswahl im ENOVIA V6 Web Interface ?

Die Type-Auswahlliste zeigt den Dictionary Type nicht an. Dementsprechend ist es auch nicht möglich im Web Interface danach zu suchen



Die Lösung für dieses Problem ist jedoch recht einfach. Die Type-Auswahl im Web Interface wird durch den Eintrag

emxFramework.GenericSearch.Types in der Datei **emxsystem.properties** konfiguriert.

Die vollständige Dokumentation zum Thema 'Search Settings' finden Sie unter:

**{Documentation
Installdirectory}/B211doc/English/DteCustomizingMap/dte-t-
searchproperties-set.htm**

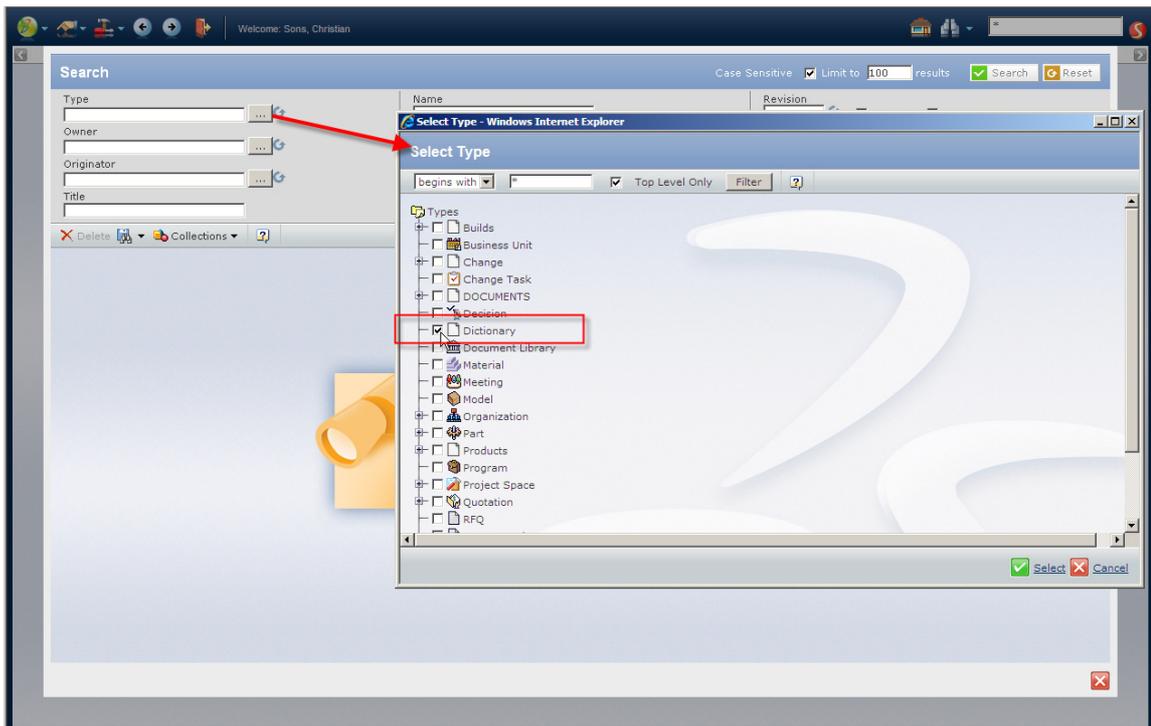
Dementsprechend ergänzen wir die Datei **emxsystem.properties** wie folgt :

Customizing ENOVIA V6 : Wörterbuch anlegen

```
# Types to display in General Search's Type Chooser  
(Inclusion List for General search)  
emxFramework.GenericSearch.Types = type_Dictionary, \  
type_Part, \  
type_SharedTable, \  
type_BusinessUnit, \  
type_Material, \  
type_Substance, \  
type_ProductData, \  
type_DOCUMENTS, \  
type_Change, \  
type_Project, \  
type_Library, \  
type_ProjectSpace, \  
type_Task, \  
type_Change, \  
type_RequestToSupplier, \  
type_Quotation, \  

```

Nach einem Tomcat Neustart und Login ist das Dictionary in der Type-Auswahlliste enthalten und Sie können nach dem zuvor erzeugten Eintrag suchen.



Nachdem Sie das 'Dictionary' ausgewählt und die Suche abgeschlossen haben,

Customizing ENOVIA V6 : Wörterbuch anlegen

The screenshot shows the search interface of ENOVIA V6. The search form includes fields for Type (Dictionary), Owner, Originator, Title, Name (MyDic*), Policy, Vault, Modified, Revision, State, and Originated. A message box with a magnifying glass icon and the text "Please enter one or more values, each one separated by a comma, into the fields provided." is displayed in the center of the page.

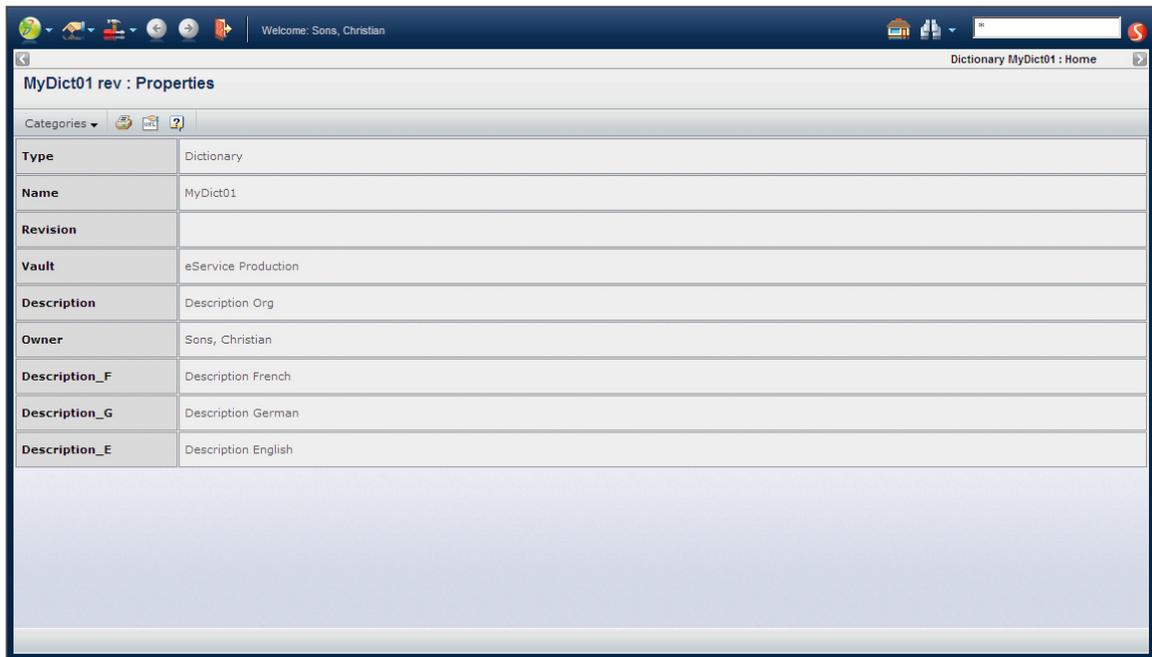
wird die Ergebnisliste in der Standard Ergebnistabelle dargestellt.

The screenshot shows the search interface of ENOVIA V6 with a search form and a table displaying the search results. The table has the following columns: Name, Revisio, Type, Description, and State. The table contains one result:

Name	Revisio	Type	Description	State
MyDict01		Dictionary	Description Org	Exists

Ebenso wird die Standard Web Form, entsprechend den allgemeinen Regeln zur Erzeugung der Web Forms, für den Wörterbucheintrag angezeigt.

Customizing ENOVIA V6 : Wörterbuch anlegen



Zusammenfassung

Das Erstellen einer angepassten Type-Struktur oder das Anpassen einer vorhandenen Type-Struktur, ist in ENOVIA V6 sehr einfach möglich. Um dem Anwender auch im Web Interface eine angepasste Suchmöglichkeit zu bieten, ist es nur notwendig, die emxsystem.properties Datei geringfügig anzupassen.

Im nächsten Post

Im nächsten Post werden Sie sehen, wie Sie das Web Interface anpassen, um einen neuen Wörterbucheintrag zu erstellen.

MQL to create the Objects

```
# Add attribute Description_E.  
#
```

```
add attribute Description_E  
  type string  
  description "Description English"  
  multiline ;
```

```
# Add attribute Description_F.  
#
```

```
add attribute Description_F  
  type string
```

Customizing ENOVIA V6 : Wörterbuch anlegen

```
description "Description French"  
multiline ;
```

```
# Add attribute Description_G.  
#
```

```
add attribute Description_G  
type string  
description "Description German"  
multiline ;
```

```
# Add type Dictionary.  
#
```

```
add type Dictionary  
description Dictionary  
abstract false  
attribute Description_E  
attribute Description_F  
attribute Description_G;
```

```
# Add policy Dictionary.  
#
```

```
add policy Dictionary  
notenforce  
description Dictionary  
type Dictionary  
state Exists  
version false  
revision false  
promote false  
checkouthistory false  
owner none  
public none;  
modify policy 'Dictionary'  
state 'Exists';
```

```
# Edit policy Dictionary.  
#
```

```
modify policy Dictionary  
state Exists add  
allstate
```

Customizing ENOVIA V6 : Wörterbuch anlegen

```
add owner all
add public all;
modify policy 'Dictionary'
state 'Exists';
```

```
# Add symbolic name
# with pure MQL like this
```

```
add property "attribute_Description_E" on program
"eServiceSchemaVariableMapping.tcl" to "attribute" "Description_E";
add property application on "attribute" "Description_E" value "Dictionary";
add property version on "attribute" "Description_E" value "1-0";
add property 'original name' on "attribute" "Description_E" value
"Description_E";
add property installer on "attribute" "Description_E" value "instDictionary";
add property 'installed date' on "attribute" "Description_E" value "15.08.2011";
```

```
# Add symbolic name
# A more convenient way to add symbolic names to ENOVIA was provided by
Dr. Ulf Landscheid from Dassault Systèmes. He uses a TCL procedure and
commit only variable dues.
```

```
tcl;
proc RegisterSymbolicName {WHAT ORIGINALNAME} {
  set SYMBOLICNAME [string map {" " ""} $ORIGINALNAME]
  set INSTALLDATE [clock format [clock seconds] -format %D]
  mql add property ${WHAT}_${SYMBOLICNAME} on program
eServiceSchemaVariableMapping.tcl to $WHAT "$ORIGINALNAME"
  mql add property application on $WHAT "$ORIGINALNAME" value "Dictionary"
  mql add property version on $WHAT "$ORIGINALNAME" value "1-0"
  mql add property installer on $WHAT "$ORIGINALNAME" value "instDictionary"
  mql add property "installed date" on $WHAT "$ORIGINALNAME" value
"$INSTALLDATE"
  mql add property "original name" on $WHAT "$ORIGINALNAME" value
"$ORIGINALNAME"
}
```

```
RegisterSymbolicName attribute "Description_F"
RegisterSymbolicName attribute "Description_G"
RegisterSymbolicName type "Dictionary"
RegisterSymbolicName policy "Dictionary"
```

```
exit
```