How to Work with OntoEdit
User's Guide for OntoEdit Version 2.6

1. Introduction ................................................................. 2
1.1. About OntoEdit ............................................................. 2
1.2. Version History ............................................................ 2
1.3. Ontologies ................................................................. 3
1.4. What is an Ontology? ................................................... 3
2. Installation ................................................................. 7
2.1. System Requirements .................................................. 7
2.2. Install Instructions ....................................................... 7
2.3. Licence File ............................................................... 7
3. Plugin Architecture ....................................................... 8
3.1.1. OntoEdit Free .......................................................... 8
3.1.2. OntoEdit ................................................................. 8
3.1.3. OntoEdit Professional ............................................. 8
3.1.4. External Plugins ..................................................... 9
3.2. Loading and Removing Plugins ................................... 9
4. User's Guide ............................................................... 11
4.1. Basic Modelling .......................................................... 11
4.1.1. Namespaces ............................................................ 11
4.1.2. Concepts & Relations .............................................. 12
4.1.3. Instances ............................................................... 15
4.1.4. Relation Axioms .................................................... 17
4.1.5. Disjoined Concepts ............................................... 17
4.1.6. Identification ......................................................... 18
4.1.7. Metadata ............................................................. 19
4.2. Advanced Modelling .................................................. 21
4.2.1. Visualizer .............................................................. 21
4.2.2. Query Tool ........................................................... 21
4.2.3. Domainlexicon ...................................................... 23
4.2.4. OntoMap .............................................................. 24
4.2.5. General axioms ..................................................... 29
4.2.6. Graphical Rule Editor .......................................... 29
4.2.7. Graphical Rule Debugger ...................................... 32
4.2.8. Inferencing ........................................................... 33
4.2.9. Data Import/Export ............................................. 35
4.2.10. Database Import/Export ...................................... 36
5. Modelling an example Ontology ..................................... 37
6. Developer's Guide ....................................................... 40
6.1. OXML 2.0 ............................................................... 40
7. Imprint .................................................................. 41

OntoEdit® is worldwide copyrighted by Ontoprise GmbH. Parts of the technology are patented or filed for patent. OntoEdit®, OntoBroker®, SemanticMiner®, OntoAnnotate®, OntoMap® and OntoCollect® are trademarks of ontoprise GmbH.
1. Introduction

1.1. About OntoEdit

OntoEdit V2.6 is an Ontology Engineering Environment supporting the development and maintenance of ontologies by using graphical means. OntoEdit is built on top of a powerful internal ontology model. This paradigm supports representation-language neutral modelling as much as possible for concepts, relations and axioms. Several graphical views onto the structures contained in the ontology support modelling the different phases of the ontology engineering cycle.

The tool is based on a flexible plug-in framework. Firstly this easily allows extending functionality in a modularized way. The plug-in interface is open to third parties which enables users to extend OntoEdit easily by additionally needed functionalities. Secondly, having a set of plug-ins available like e.g. a domain lexicon, an inferencing plug-in and several export and import plug-ins, this allows for user-friendly customization to adapt the tool to different usage scenarios.

OntoEdit V2.6 is available in a free and a professional version. The professional version includes an additional set of plug-ins, e.g. the inferencing capabilities. Currently version 2.6 is available, 2.7 is scheduled for 2003-02.

1.2. Version History

**OntoEdit 2.0**  Release Date: 01.01.2002

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts Modeling</td>
<td>Creating and editing of concepts, building of taxonomies</td>
</tr>
<tr>
<td>Disjoint concepts</td>
<td>Constraint Modeling: Disjoint concepts</td>
</tr>
<tr>
<td>F-Logic support</td>
<td>Support of the logic format F-Logic, import/export-filter</td>
</tr>
<tr>
<td>Metadata administration</td>
<td>entering of metadata concerning the ontology (URI etc.), statistics (number of concepts, depth)</td>
</tr>
<tr>
<td>Ontology Identification</td>
<td>saving of metadata concerning the ontology (sources, developer, etc.)</td>
</tr>
<tr>
<td>RDF support</td>
<td>support of the data format RDF</td>
</tr>
<tr>
<td>Relation Axioms</td>
<td>Modeling symmetry, inversity and transitivity</td>
</tr>
<tr>
<td>Textual Rule Editor</td>
<td>Textual rule builder and editor</td>
</tr>
</tbody>
</table>

**OntoEdit 2.5**  Release Date: 01.05.2002

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAML support</td>
<td>import and export out of DAML format</td>
</tr>
<tr>
<td>Database integration</td>
<td>Import and export of database schemes and facts, supported databases: Oracle, MSSQL, DB2</td>
</tr>
<tr>
<td>Domainlexicon</td>
<td>Defining of synonyms and related terms</td>
</tr>
<tr>
<td>instance editing</td>
<td>Entering and editing facts (instances and their attributes)</td>
</tr>
<tr>
<td>Integrated Inferencing</td>
<td>a light-weighted inferencing mechanism for testing the rules</td>
</tr>
<tr>
<td>multilingualism</td>
<td>defining external representations including multilingualism</td>
</tr>
<tr>
<td>Querying the ontology</td>
<td>Querying for facts contained in the ontology or in related data sources (i.e. databases)</td>
</tr>
<tr>
<td>Relation Modeling</td>
<td>Creating and editing of relations, types of relations, 1-n, etc</td>
</tr>
<tr>
<td>Search for concepts</td>
<td>Concept Search in the tree</td>
</tr>
</tbody>
</table>
1.3. Ontologies

Ontologies serve as a means for establishing a conceptually concise basis for communicating knowledge for many purposes. We restrict our attention to domain ontologies, that describe a particular small model of the world as relevant to applications and have shown their usefulness in application areas such as intelligent information integration, information brokering, natural-language processing, metadata generation, knowledge management, to name but a few.

Typically, an ontology is constructed and maintained in a collaborative effort of domain experts, end-users and IT specialists. OntoEdit is a tool which enables inspecting, browsing, codifying and modifying ontologies and supports in this way the ontology development and maintenance task.

Modeling ontologies using OntoEdit means modeling at a conceptual level, that means as much as possible independence of a concrete representation language and using GUI's representing views on conceptual structures (concepts, concept hierarchy, relations, axioms) rather than codifying conceptual structures in ASCII. The conceptual model of an ontology is internally stored using a powerful ontology model, which can mapped onto different, concrete representation languages.

1.4. What is an Ontology?

An ontology is an advanced knowledge representation model. Other models e.g. are database schema, thesauri or topic maps. The most common definition of “ontology” is “An ontology is an explicit specification of a conceptualization.” by Tom Gruber, another is:

“An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base.” by Swartout, Patil, Knight and Russ.
OntoEdit Tutorial

An Ontology consists of several components, the most important are:

- Concepts
- Relations and Attributes
- Instances
- Axioms

**Concepts** are abstract terms that are organized in taxonomies. Hierarchical concepts are linked with an "is-a" relation.

```
http://manual.de (New ontology)
```

![Concept hierarchy diagram]

e.g.: 2 concepts: PERSON, WOMAN. "A woman is a person."

**Relations** link non-hierarchical concepts together.

```
New relation of company
```

![Relation diagram]

e.g.: 2 concepts: COMPANY, PERSON, 1 relation: “works in”. “A company occupies a person.”

**Attributes** are relations of pre-defined data types like STRING, INTEGER, BOOLEAN etc.
e.g.: 1 concept: PERSON, 1 relation: HASGIVENNAME (of type STRING). “A person has a name.” Inheritance is also provided in ontologies: Also a WOMAN has a name, because she’s a subconcept of PERSON.

Instances are concrete occurrences of abstract concepts.

e.g.: 1 concept: MAN, 1 Instance: DIRKWENKE. “DirkWenke is a man. His first name is Dirk.”
Axioms
Axioms are rules that are valid in the modelled domain. There are simple symmetric, inverse or transitive axioms and complex rules consisting of several relations. e.g. an inverse axiom: If a person works in a company, the company employs this person.
2. Installation

2.1. System Requirements
To be able to run OntoEdit you should match the following system requirements:

CPU : Pentium II (400 MHz)
RAM : 128 MB
At least 20 MB free Hard Disc space
Java Runtime Environment JRE 1.4

2.2. Install Instructions

Windows
To install OntoEdit you have to start the OntoSetup.exe which will launch a setup wizard which will guide you through the installation process. After this, you can start OntoEdit by clicking on "Start" → "Programs" → "Ontoprise" → "OntoEdit. This may take a few seconds, because the ontoprise Update Service is checking for available software updates of OntoOffice on our web server before launching the application.

Other platforms
Before installing OntoEdit you have to install the Java Runtime Environment JRE 1.4. To install OntoEdit you have to extract the OntoEdit.zip file to the desired OntoEdit directory. Then you have to add the included .jar files to the CLASSPATH variable. Now you can start OntoEdit by typing:

java com.ontoprise.oee.gui.OntoEdit

2.3. Licence File
If you purchased a non-free edition of OntoEdit, you have to copy the given licence file into the OntoEdit root folder (i.e. ../program files/Ontoprise/OntoEdit/).
OntoEdit Tutorial

3. Plugin Architecture

OntoEdit 2.6 is based on an open plugin structure. Every plugin provides other features to deal with the requirements an ontology engineer has. So the plugin concept allows the user to select individually only the plugins he needs.

Ontoprise offers its OntoEdit plugins in three different packages:

- OntoEdit Free
- OntoEdit
- OntoEdit Professional

Developers are invited to build their own OntoEdit plugins by using our plugin tutorial.

3.1.1. OntoEdit Free

OntoEdit Free contains the basic features needed to model an ontology. Data import and export filters to standard formats like RDF or DAML are also provided as the visualisation of the model.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept hierarchy, Attributes, Relations</td>
<td>Base</td>
</tr>
<tr>
<td>Multilingualism</td>
<td>Base</td>
</tr>
<tr>
<td>Multiple Ontology Management</td>
<td>Base</td>
</tr>
<tr>
<td>RDF(S)</td>
<td>Export &amp; Import filter</td>
</tr>
<tr>
<td>OXML</td>
<td>Export &amp; Import filter</td>
</tr>
<tr>
<td>DAML+Oil</td>
<td>Export &amp; Import filter</td>
</tr>
<tr>
<td>F-Logic</td>
<td>Export &amp; Import filter</td>
</tr>
<tr>
<td>Excel</td>
<td>Import filter</td>
</tr>
<tr>
<td>Directory structures</td>
<td>Import filter</td>
</tr>
<tr>
<td>Instance editor</td>
<td>Plug-In</td>
</tr>
<tr>
<td>Disjoined concepts</td>
<td>Plug-In</td>
</tr>
<tr>
<td>Visualizer</td>
<td>Plug-In</td>
</tr>
</tbody>
</table>

OntoEdit Free is limited to the management of 50 concepts, 50 relations and 50 instances.

3.1.2. OntoEdit

OntoEdit contains all features of OntoEdit Free plus the Domainlexicon plugin and has no limitation to the management of concepts or relations.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoEdit Free</td>
<td>Base</td>
</tr>
<tr>
<td>Domainlexicon</td>
<td>Plug-In</td>
</tr>
<tr>
<td>No limitation to the management of concepts or relations.</td>
<td></td>
</tr>
</tbody>
</table>

3.1.3. OntoEdit Professional

OntoEdit Professional is based on OntoEdit and enables the user additionally to build rules, querying and inferencing, import from database schema and export into
OntoEdit Tutorial

3.1.4. External Plugins

Information on Plugins by external suppliers can be found at http://www.ontoprise.de/products/ontoedit_plugins_en.

3.2. Loading and Removing Plugins

The OntoEdit plugin architecture is hot-pluggable. That means that you can load and unload plugins during runtime. The plugins can be managed in the "Manage Tools" dialog. This dialog is accessible via Tools > Manage...

On the left side of the dialog the already known plugins are listed. On the right side the installed plugins are shown divided into three types: Import/Export plugins, FramePanel plugins and other plugins.

**Loading a new plugin**

If you have a new plugin (developed or purchased) you have to put the classes in the OntoEdit installation directory. If you have a .jar file with your classes, you have two possibilities:

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoEdit</td>
<td>Base</td>
</tr>
<tr>
<td>Textual Rule Editor</td>
<td>Plug-In</td>
</tr>
<tr>
<td>Graphical Rule Editor</td>
<td>Plug-In</td>
</tr>
<tr>
<td>Graphical Rule Debugger</td>
<td>Plug-In</td>
</tr>
<tr>
<td>SQL-Schema</td>
<td>Export filter</td>
</tr>
<tr>
<td>SQL-Schema</td>
<td>Import filter</td>
</tr>
<tr>
<td>Basic axioms</td>
<td>Plug-In</td>
</tr>
<tr>
<td>Inferencing</td>
<td>Plug-In</td>
</tr>
<tr>
<td>Query Tool</td>
<td>Plug-In</td>
</tr>
<tr>
<td>OntoMap</td>
<td>Plug-In</td>
</tr>
</tbody>
</table>
OntoEdit Tutorial

- extract the classes of the .jar file into the OntoEdit installation directory, or
- add the .jar file to the CLASSPATH variable. The problem in this case is that the classpath cannot be changed during runtime, so you first have to exit OntoEdit, add the file to the classpath and then restart OntoEdit.

Now you can add the new plugin by clicking the "Add" button. A dialog box appears where you have to enter the name of the class of the new plugin (complete name with path, like "com.ontoprise.oee.gui.InstanceEditorPanel"). If the name is correct, the plugin will appear on the left side of the dialog shown above.

Install a loaded plugin
You can install a loaded plugin by selecting it and then pressing the "->" button. After pressing the 'ok' button, the newly installed plugins will be initialized.

Uninstall a plugin
You can uninstall a plugin by selecting it and then pressing the "<-" button. After pressing the 'ok' button, the plugin will be uninstalled. The plugin will remain in the list of available plugins.

Unload a plugin
If you want to remove a plugin completely from the environment, you can select it and then press the "Remove" button. It will also be removed from the list of available plugins.

Plugin info
To get information about the plugins, you can select them and press the info button.
4. User’s Guide

4.1. Basic Modelling

OntoEdit V2.6 is delivered with many standard plugins. First of all there are 6 FramePanel plugins each displayed by a tab in the Ontology Frame:

- Concepts & Relations
- Instances
- Relation axioms
- Disjoint concepts
- Identification
- Metadata

To switch between the documentation of the different panels you can click on the tabs of the panels.

4.1.1. Namespaces

When creating a new ontology, you will be asked for a namespace resp. a URI (Uniform Resource Identifier), that will allocate a concept to an ontology.

If you want to define an own standard namespace for your ontology, please type it into this field.

Select the item "View..Show namespaces" on the toolbar to show the namespaces in
the concept hierarchy. If you want to integrate concepts from other ontologies directly into your ontology, you can do this by changing the namespace to the one of the other ontology.

4.1.2. Concepts & Relations

In the Concepts & Relations Plugin you can insert, edit and remove concepts and relations. The concepts are arranged in a is-a-hierarchy in the tree on the left side. If you select a concept in the hierarchy you can see its relations and their ranges on the right side.

Inserting concepts
To insert a (sub-)concept you first have to select a concept which should be the superconcept of the new concept. Now you can insert a new concept by either pressing the “+” button or by selecting “Insert concept” from the context menu reachable by a right mouse-click. A new item in the tree appears where you can change the concept's name. The name of the concept has to meet some restrictions. Its characters may only be letters, numbers and ‘_’(underscore) where the first character has to be a letter or an underscore.

Removing concepts
To remove a concept from its superconcept you have to select the concept which should be removed. Then you can remove the concept by either pressing the “-” button or by selecting “Remove concept” from the context menu.

Editing concepts
To edit a concept you first have to select the concept. Now you can edit the concept by either pressing the “e” button or by selecting “Edit concept” from the context menu. The following dialog box will appear:
OntoEdit Tutorial

In this dialog box the user can change the concept ID, set the instantiation and define external representations and documentation. Instantiation can be set to 'abstract' or 'concrete'. 'Abstract' means that this concept cannot be instantiated, 'concrete' allows instantiation. The external representation can be defined for different languages. For each language there is only one external representation which is the most commonly used word/synonym for the concept. Adding, removing and editing external representations works via the '+' , '-' and 'e' buttons. Finally a documentation can be defined for each language.

Reorganizing concepts
Reorganizing means copy or move in this case. You can 'reorganize' either by Drag&Drop or by a right click. To copy a concept by Drag&Drop, you must press the Ctrl-button. If you select a concept and right-click in the window, you can select 'Reorganize' from the popup menu and choose between 'copy' and 'move'. After selecting the operation you can choose the new superconcept in a list of all concepts. 'Move' means removing the concept from the actual superconcept and inserting it as subconcept of the new selected superconcept. 'Copy' means only inserting the concept as subconcept of the new selected superconcept. This is a possibility of gaining multiple inheritance.

Inserting relations of concepts
First you have to select a concept. To insert a new relation of the selected concept, you can either right-click in the concept tree and choose 'insert relation' or right-click in the relation area and choose 'insert relation'. A dialog box will appear.
In this dialog the relation ID can be given, the range can be specified and the cardinalities. As already known from the concepts external representations and documentation can be defined. The range of a relation can either be a built-in concept (STRING, BOOLEAN or INTEGER) or a set of new datatypes (CONCEPTS, RELATIONS or FILE) or another concept of the ontology.

Removing relations from concepts
To remove a relation of a concept you have to select the desired concept, then select the relation to remove and select from the popup menu 'Remove relation'.

Editing relations of concepts
To edit a relation, select the relation and choose 'edit relation' from the context menu. The same dialog as used for the creation of a new relation will appear.
4.1.3. Instances

In the instances plugin you can define, edit and remove instances (the facts according to the schema consisting of concepts and relations). On the left side you can see the concept hierarchy and on the right side all instances of the currently selected concept are listed. Because the concept hierarchy is an 'is-a' hierarchy, all instances are also instances of all superconcepts. That means e.g. that all instances of 'man' are also instances of 'person' because a man is a person.

**Defining instances**

To define a new instance you have to select the concept you want to instantiate. Then right-click in the instances area and select 'add instance' from the popup menu. A new instance will be inserted and the user is asked to change its identifier.

**Editing instances**

To edit an instance you have to select the desired instance and choose 'edit instance' from the popup menu. The following dialog will appear.
In the dialog you can change the identifier of the instance and set values of relations. In the left column of the table you can click a row and choose a relation you want to fill with information. Thereafter you can enter this information in the left column. If the range of the relation is a concept, you will get a list of all instances of the range concept. If the range is a built-in like 'STRING', etc. you can edit the table cell and enter the value.

You can also change the instantiated concept, but this might lead to the loss of some relation values, if the used relations do not exist for the new concept.

**Removing instances**
To remove an instance you have to select the desired instance and choose 'remove' from the popup menu. You can also only remove a relation/value pair. To do this you have to click the handle of the instance and all the relation/value pairs will be shown. Selected the pair you want to delete and choose 'remove' from the popup.
4.1.4. Relation Axioms

In the relation axioms plugin some predefined axioms can be defined. Currently the following axioms are available:

- Symmetric axioms
- Transitive axioms
- Inverse axioms

Adding a new axiom
To define a new axiom, you have to select the type of axiom first. This can be done via the combo box in the upper right corner. Now you can define an axiom by selecting the relation you want to be e.g. symmetric and then press the '+' button.

Editing an axiom
You can edit an axiom by clicking in the row of the table. If you select a cell in the table a combo box will appear with possible values.

Removing an axiom
To remove an axiom simply click in the associated row of the table and click the '-' button.

4.1.5. Disjoined Concepts

In this plugin you can define disjoined concepts. This is not really an axiom, but a constraint that can be used to check the consistency of the ontology. If two concepts are disjoint, there cannot exist instances that instantiate both concepts. Right now OntoEdit allows only instances that instantiate one concept, but in future it will be allowed to instantiate multiple concepts.

Defining disjoint concepts
To define disjoint concepts, you can select all concepts you want to be disjoint (hold <shift> or <ctrl> for multiselection) and then press the '+' button. Pairwise disjointness will be assumed if more than two concepts are selected. That means that all permutations will be computed.
Changing axioms
To change the concepts that are defined as disjoint, you can simply click in the table. A combo box will appear where you can change the selected concept.

Removing axioms
To remove a pair of disjoint concepts, you can select the row and then press the '-' button.

4.1.6. Identification

In the identification plugin some meta information about the ontology can be specified. Here you can give the ontology a title, define the domain and the application area. The URI (Uniform Resource Identifier) is shown here, but cannot be edited. You can also specify related ontologies (that may be ontologies that have been used to develop this
ontology or ontologies of the same domain or application area) and sources (books, papers, etc. used to built this ontology).

**Adding a new related ontology**
To add a related ontology, you can simply enter the location or the name of the ontology in the free row of the table. If you finish editing, a new empty row will be added.

**Removing a related ontology**
You can remove a related ontology by selecting the row of the table and choosing 'remove' from the popup menu (after right mouse-click).

**Adding a new source**
To add a new source, right-click in the list of sources and select 'new source' or press the '+' button. A new item will appear in the list of sources and you can edit this entry in the textarea below. If you finish editing, you will be asked if you want to store the changes.

**Editing a source**
To edit a source, select it in the list and then edit the source in the textarea below. If the textarea loses the keyboard focus, you will be asked to store the information.

**Removing a source**
To remove a source, select it in the list and choose 'remove' from the popup menu.

4.1.7. Metadata
In this plugin you can see statistic information of the ontology and specify the developers of the ontology and documentation for all languages. The statistics contain the number of concepts, relations, instances and axioms, the highest depth and the average depth in the concept tree, the creation date and the date of the last modification.

**Adding a developer**
To add a new developer to the list of developers right-click in the list of developers and select ‘add developer’ from the popup menu. A new developer will be added to the list and the user can change the data of the developer. If the editable fields lose the keyboard focus, the user will be asked if he wants to store the changes.

**Editing a developer**
To edit the data of a developer you have to select the developer in the list of developers. Now you can edit the developer data and you will be asked to store the changed data if the editable fields lose the keyboard focus.

**Removing a developer**
To remove a developer from the list, simply right-click in the list of developers and select ‘remove developer’ from the popup menu.

**Adding documentation**
To add a documentation you have to simply right-click in the table of the documentations and select ‘add documentation’ from the popup menu. Now you can select the documentation language and write the documentation. If the editable fields lose the keyboard focus, you will be asked to store the documentation.

**Editing documentation**
To edit a documentation you have to select the desired documentation from the table and then you can edit it. The changed documentation will be stored after a change of the keyboard focus.

**Removing documentation**
To remove a documentation you have to select the documentation in the table and then right-click and select ‘remove documentation’.
4.2. Advanced Modelling

4.2.1. Visualizer

The Visualizer plugin allows the user to browse and edit the ontology by a graphical way.

**Navigate**

Concepts and Instances are shown as blue and green bullets, which can be used as navigators either by doubleclick or right-click. The number of existing instances to a selected concept is shown in a small red square.

**Edit**

To edit an ontology, you have to select the 'Edit' radio button. The right click menu on the bullets shows you the options for editing the ontology.

4.2.2. Query Tool

The Query Tool is a plugin which helps to query the knowledge base. It enables the user to generate the queries graphically based on one of the available ontologies in OntoEdit V2.6.

**Generate a query on one concept**

The easiest query you can generate with the Query Tool is a query on one concept, e.g. display all names and email addresses of persons. The starting point of the formulation of a query is a concept in the concept hierarchy tree. First of all switch to those concept in the tree, and double click on it. A new window opens showing all attributes, form fields for these attributes, all relations of the concept and the concepts the selected concept is related to via these relations.
Now the fields may be filled to query for instances of the selected concept with the specified attributes value and an operation from the pull down menu on the left of the form field has to be selected. There are the following options for operation to select:

<table>
<thead>
<tr>
<th>Sign</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>The attribute value in the text field has to be <strong>exactly the same</strong> as the attribute value of the instance.</td>
</tr>
<tr>
<td>&gt;</td>
<td>The attribute value of the instance has to be <strong>greater than</strong> the value filled in the text field.</td>
</tr>
<tr>
<td>&lt;</td>
<td>The attribute value of the instance has to be <strong>less than</strong> the value filled in the text field.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>The attribute value of the instance has to be <strong>greater or equal than</strong> the value filled in the text field.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>The attribute value of the instance has to be <strong>less or equal than</strong> the value filled in the text field.</td>
</tr>
<tr>
<td>Contains</td>
<td>The attribute value of the instance has to <strong>contain the substring</strong> filled in the text field.</td>
</tr>
<tr>
<td>startsWith</td>
<td>The attribute value of the instance has to <strong>start with the substring</strong> filled in the text field.</td>
</tr>
<tr>
<td>endsWith</td>
<td>The attribute value of the instance has to <strong>end with the substring</strong> filled in the text field.</td>
</tr>
</tbody>
</table>

Leaving such a text field blank asks for all values for this attribute. To show an attribute in the result, you have to select the check box *show*, else all your settings you made in this row will not be used. It is also possible to create a query which shows you all attribute values which are not equal than a specified value, by selecting the *not-* check box.

If all settings in the window with the attributes are made, don’t close it. In this case, all settings you made will be resetted. Click on the *Generate* Button in the main window of the Query Tool.

The tool will generate a query from all the settings you made and show the results.

**Generate a query on more than one concept**

You also have the possibility to generate queries on more than one concept, what means, that to concepts are linked with each other, e.g. *all names of persons and the title of the documents they have written*. You already learned how to generate a query...
on one concept. If the concept you opened has relations to other concepts, they are
listed in the window with the attributes and relations.

![Attributes and Relations of Concept: Person](image)

By pressing the button on the right of the relation name you open a window with the
attributes and relations of the related concept. If you select a value in this window, the
first concept you opened via the concept tree and the second window you opened by
the button are linked via the relation, e.g. person → hasWritten → document. This has
the effect, that all persons listed in the results must be an author of a document.

4.2.3. Domainlexicon

The domainlexicon plugin enables the user to expand the ontology by additional
expressions of the concepts. These expressions can be:

- synonyms (same meaning as the selected concept)
- domain entries (same domain as the selected concept)
- classifications (documents belonging to the selected concept)

Multilingualism is provided by the selection field "Language".
Attribution
You can attribute a synonym to a concept by typing it in the left column. After that you can select the language and the type of the attribution (synonym, domainentry or classification.

Sorting and Filtering
By clicking on the headline buttons the table will be sorted by the selected criterion. You can set filters by selecting the three drop-down-fields above.

Import/Export
You can import and export Domainlexicons into OXML or F-Logic format. The SemanticMiner import and export suits to ontoprise's knowledge retrieval platform SemanticMiner.

4.2.4. OntoMap
The OntoMap plugin allows the mapping of data structures between ontologies. Open the two ontologies which you want to map. With the menu bar entry Tools..OntoMap..show/hide you will open the mapping window.
By the dropdown-lists you can select the two ontologies:

mapping rules

You have several mapping possibilities:

- concepts on concepts
- attributes on concepts
- attributes on attributes
- relations on relations

1. concepts on concepts
Draw a concept by Drag&Drop from one side to the other → there appears a blue arrow.
2. attributes on concepts

For database integration this mapping is very important. Draw a connection from an attribute on one side to a concept on the other side. It appears a violet arrow for attribute-concept-mapping.

3. attributes on attributes

Before connecting two attributes, you have to map their concepts first (1.), or there's already an attribute-to-concept mapping (3.). In this case you can connect these attributes, it appears a red arrow.
4. relations on relations

Note: attributes and relations are both marked by the - icon and won’t distinguished further. Relations link concepts to each other (i.e. Person works_for Company), attributes allocate ranges to concepts (i.e. name of Person is of range STRING).

To map relations-to-relations, you have to map all allocated concepts first (here Company and entreprise). So draw first all required concepts-to-concepts-mappings and afterwards the relation-to-relation-mapping.
4.2.5. General axioms

The General Axiom plugin allows the user to define all kinds of axioms. On the left side of the window all existing axioms are shown, the predefined axioms (inverse, disjoint, etc.) in their own folders and also the general axioms named 'flogic_axiom1', 'flogic_axiom2', etc. The syntax of the axioms has to be F-Logic (Frame Logic). To get more information about F-Logic you can download an F-Logic tutorial from our website.

Defining a new axiom
To define a new axiom you can right-click in the axiom tree and select 'new axiom' from the popup menu. Thereafter you can type the syntax into the upper textarea on the right side. You can also give the axiom a documentation by pressing the '+' button.

Editing an axiom
To edit an axiom, you have to select it and the axiom information will be shown. Now you can change the syntax or the documentations. You can also define documentations for the predefined axioms.

Removing axiom
To remove an axiom you have to select it and then choose 'remove axiom' from the popup menu (right mouse-click). You can also remove the predefined axioms.

4.2.6. Graphical Rule Editor

The Graphical Rule Editor plugin allows you to create rules. Unlike the General Axioms plugin (textual rule editor), you are assisted by a graphical interface, where no F-Logic knowledge is needed.
OntoEdit Tutorial

In our showcase we will build a simple rule: "If a Person works in a Project, and in this Project there is required a special Knowledge, then this Person has this Knowledge."

In F-Logic, this rule would have the following syntax:

\[
\text{FORALL Person1,Project1,Knowledge1 ( Person1[\#has_knowledge_in->>Knowledge1] )} \\
\quad \leftarrow ( \text{Person1[#name->>name1;#works_in->>Project1} \\
\quad \text{and Project1[#requires_knowledge_in->>Knowledge1] \\
\quad \text{and Knowledge1[#Knowledge]})}. \\
\]

As you see, a rule consists of two parts: the section for the rule heads (before '<-') and the section for the rule bodies (after '<-').

Defining the rule body
At first we have to define the rule body: "If a Person works in a Project, and in this Project there is required a special Knowledge".
At first, you click on one concept, which is included in the rule body. In this example it's the concept **Person**. Because a Person works in a Project, you have to click the button named with the related concept **Project**. (picture 1) Because a Project requires Knowledge, you then have to click the button **Knowledge** in the new tab. (picture 2)

**Defining the rule head**
Now we will define the rule body: ".. then this **Person** has this **Knowledge**." Therefore we select the variable **Person1** from the field 'Use Variable'. (picture 4)
After enlarging the rule head sector we select Knowledge1 from the selection box in the referring row has_knowledge_in. By clicking the 'generate'-Button the rule will be created.

**Editing/Removing axiom**
To edit or remove an axiom you have to select it in the 'graphicalaxioms' folder and then choose 'edit axiom' or 'remove axiom' from the popup menu (right mouse-click).

### 4.2.7. Graphical Rule Debugger

The Graphical Rule Debugger visualizes the inferencing process activated by a query. Select the “Debugger” tab in OntoEdit and click the “Refresh” button. As a result, the inference graph of the ontology will appear. The rules (axioms) are displayed by the “A” icon, the facts by the “C” icon, the black circle mark the facts that are sent to the axioms:
Enter a query into the appropriate field, and the debugger will display what the inference engine executed.

Continuing the former example in the rule editor chapter, the query \( \text{FORALL } X,Y \leftarrow X[\#\text{has_knowledge} \_\text{in} \rightarrow Y] \) would throw out the inferred result \( X = \text{Hans}, \ Y = \text{knowledge1} \).

The incoming facts needed for this inference are displayed by a mouse-over on the appropriate facts element:

The output generated by the axiom is displayed by a mouse-over on the appropriate axioms element:

4.2.8. Inferencing

The inferencing plugin can be used to test the ontology and its axioms. In the text field
on the upper right you can type queries to query the data model. These queries have to be in F-Logic syntax. To get more information about F-Logic you can download an F-Logic tutorial from our website.

Another functionality is the possibility of enabling and disabling axioms. This is the way of testing axioms. A simple example:
You have modeled the two concepts ‘person’ and ‘company’. ‘person’ has a relation ‘worksFor’ to ‘company’ and ‘company’ has a relation ‘occupies’ to ‘person’. You also defined ‘DirkWenke’ as an instance of ‘person’ and ‘ontoprise’ as an instance of ‘company’. You edited ‘DirkWenke’ and filled the relation ‘worksFor’ with the instance ‘ontoprise’.

If you now ask for all people occupied by ontoprise
(query : FORALL X <- ontoprise[occupies->>X]. )
you will get no result. If you defined the relations ‘occupies’ and ‘worksFor’ to be inverse, you have the possibility to enable and disable this axiom. Disabling means that the axiom will not be used for the query. The result of the query will be the same as before. But if you enable the axiom, you will get ‘DirkWenke’ as result. This is the way to test single axioms or even sets of axioms.
4.2.9. Data Import/Export

OntoEdit V2.6 supports the data formats RDF(S), DAML+OIL, OXML and FrameLogic. Import or export the files over the menu bar and "File..Import/Export.". To integrate databases use the database import/export.

Excel import
Save your Excel table as txt-File (tab-stop-separated). Then you can import it into OntoEdit over the menu bar and "File..Import/Export.". An Excel table has to fulfil some conditions to be imported into OntoEdit:

- The first two lines must contain the data schema information like shown below. First the name of the concept (A1), then the names of its relations (B1-E1) and their range (B2-E2).
- Beginning with line 3 there are the instances. Column A contains the instance ID and can be left empty. Note that an instance ID must not occur twice. The following Columns (B-E) show the values of the instance.
- Namespaces: If there's no namespace given (i.e. http://www.a.com/#), the standard namespace will be inserted.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
</table>
| 1   | Person | lastname | telefonnummer | elBetreten | elBeliebter | ...
| 2   | STRING | STRING  | Projekt    | Firma  |...|
| 3   | person1 | Jens Muller | 0721 7921901 | BIZON | ontoprise |...
| 4   | person2 | Ulrike Schneider | 06543 230987 | OntoScrape | ontoprise |
| 5   |       |          |            |        |         |
| 6   |       |          |            |        |         |
| 7   |       |          |            |        |         |
| 8   |       |          |            |        |         |
| 9   |       |          |            |        |         |
| 10  |       |          |            |        |         |

Directory import
You can import the names of folder directories by selecting "File..Import/Export..Directory Import".

XBRL import
You can import XML-Files supporting the XBRL standard for business reportings by selecting "File..Import/Export..XBRL Import".
4.2.10. Database Import/Export

Export your ontology into a database or import schema information from a database in OntoEdit with the database import/export plugin. Select the menu bar item "Import..from database" or "Export..into database" to get a dialog shown below.

The database import/export plugin supports all databases with a JDBC interface, e.g. MSSQL Server, Oracle, DB2, MySQL, etc.

Note: When using MSSQL Server, please use the “mixed mode”. Mixed mode means that both the Windows and the SQLServer authentification will be enabled. You can adjust it in your SQL Server using the Enterprise Manager, selecting the computer, right click on “Properties” and selecting the Panel “Security”.
5. Modelling an example Ontology

This short example shows the common functionalities of OntoEdit and gives the user a short overview over the ontology engineering process.

The first step is to open a new ontology. A dialog box will appear where the URI (Uniform Ressource Identifier) has to be given. The URI has to be a web address which must start with an internet protocol (like 'http://' or 'ftp://'). This should be the address to access the ontology. A frame for the new ontology will appear.

The standard procedure of ontology development starts with the development of the concept hierarchy. With OntoEdit you can model ontologies in a 'is-a' hierarchy. That means that subconcepts have all properties of their superconcepts. If you model e.g. 'man' as subconcept of 'person' it will have all properties of person because 'man' 'is-a' 'person'.

To insert a new concept you have to select the superconcept, click the right mouse button and select 'insert concept' (or press the '+' button).
A new concept with a generic identifier will be inserted as subconcept of the selected concept. This identifier can be changed by the user by either clicking on the selected concept, choosing 'edit concept' from the context menu or pressing the 'e' button. We insert 'person' and 'company' as subconcepts of 'Root' (the predefined root of the concept hierarchy) and 'man' and 'woman' as subconcepts of 'person'.

After the development of the concept hierarchy the definition of relations/attributes is the next step. To define a relation of a concept you have to select the desired concept and then select 'insert relation' of the context menu. The following dialog box will appear.

In this dialog box the identifier of the relation can be defined, its range (which can either be a concept or a datatype) and the cardinalities. You can also define external representations and documentations. You can add, remove and edit external representations and documentations with the '+', '-', and 'e' buttons.
6. Developer’s Guide

6.1. OXML 2.0

7. Imprint

Editor
ontoprise GmbH
Amalienbadstraße 36
(Raumfabrik 29)
76227 Karlsruhe (Germany)
Telefon    +49 (0) 721 / 509 809 10
Telefax    +49 (0) 721 / 509 809 11
Email       support@ontoprise.de
Internet    http://www.ontoprise.de

© by ontoprise GmbH, all rights reserved


Karlsruhe, April 2003