

run

010010101011101100101
001010111010101010011
00110100100111001010
11010110101001011101
11000101110101010101
11011001010010101101
01010100110011010010
01110010101101011010
10010111011100010111
01010101011101100101
00101011010101010011
00110100100111001010
11010110101001011101
11000101110101010101
11011001010010101101
01010100110011010010
01110010101101011010
10010111011100010111
01010101011101100101
00101011010101010011
00110100100111001010
11010110101001011101
11000101110101010101
11011001010010101101
01010100110



run Software-Werkstatt GmbH
Weigandufer 45
12059 Berlin

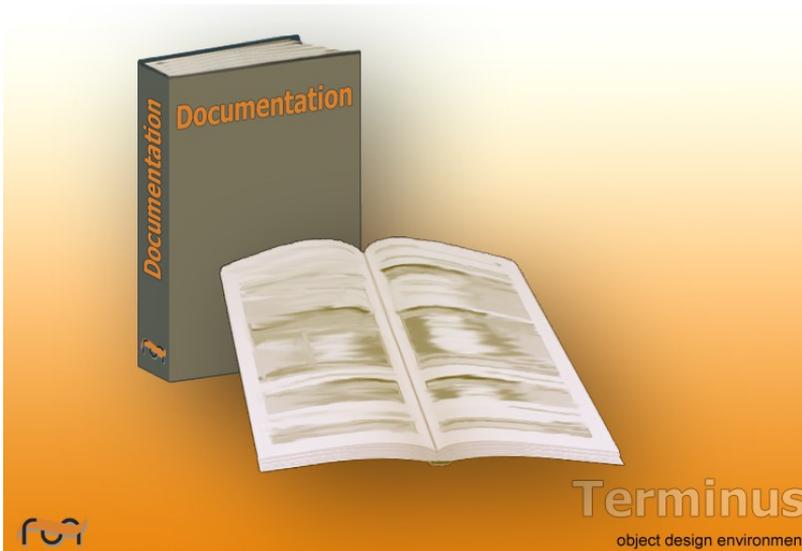
Tel: +49 (30) 609 853 44
e-mail: run@run-software.com
web: www.run-software.com

Berlin, October 2012

Content

1	Introduction	5
2	Running Terminus	7
3	Overview	9
4	Developing a terminology model	13
5	Terminus Tools	14
5.1	Global Terminus tools	15
5.1.1	Translation support	16
5.1.2	Index services	17
5.1.3	Text search	19

Terminus



Terminus is a tool, which supports defining simple concept systems, but also complex terminology models and hierarchically structured documentation.

A concept system consist of several concept definitions and links between those. The Terminus concept system provides a subset of the ISO 1087 standard for concept systems [1087].

The terminology model is a more specific definition of subject expert knowledge. The terminology provides the structure for storing information about a subject matter area properly. Since the terminology model defines knowledge elements in terms of subject area experts, but in the structure defined by the terminology model, which supports requirements of a terminology database [TMDB], the terminology model specification is a mean of common understanding and implementation guideline.

Hierarchical topic structures allow composing topics in different hierarchies. A topic hierarchy is similar to a chapter hierarchy provided in a document, but it is more flexible, since topics might be referenced in different hierarchies, and different types of output might be generated.

In order to get an overview about existing topics, a list of topics is provided as well. Thus, experienced users may enter or update topics directly in the topic list.

1 Introduction

ODABA

ODABA is an terminology-oriented database system that allows storing objects and methods as well as causalities. As terminology-oriented database, ODABA supports complex object types (user-defined data types) defined in a terminology model, which are reflect application relevant concepts.

ODABA-applications are characterized by high flexibility. In addition to object type or context hierarchies, ODABA supports multifarious relations between object instances (master and detail relations, relations between independent object instances and others). This way, behavior of objects in the real world can be represented considerably better than in relational or object-oriented systems.

ODABA supports event-driven applications concerning the graphical user interface as well as the database level. Thus, application design is very close to the experts or customers problem, since it still refers to the same names and concepts. This enables ODABA to solve highly complex jobs in administrative and knowledge areas.

Platforms

ODABA2 supports windows platforms (Windows95/98/Me, Windows NT and Windows 2000) as well as UNIX platforms (Linux, Solaris).

You can build local applications or client server applications with a network of servers and clients.

Interfaces

ODABA2 supports several technical interfaces:

- C++, .Net as application program interface (this allows e.g. using ODABA in C# or VB scripts and applications)
- ODABA Script Interface (OSI) for accessing data via a script language, which is similar to C# or JAVA.
- Multiple storage support for using relational databases for storing ODABA data
- XML for supporting data exchange with complex data structures
- OIF (object interchange format), flat files and ESDF (extended self delimiter fields) for accessing data provided in external file formats
- Document exchange support for importing or exporting data from/to open office or Microsoft office documents.

Tools

ODABA provides a number of database maintenance tools, but also development tools in order to provide terminology model definitions, data model specifications, application design and others.

In order to support just-in-time documentation, all ODABA tools provide extended documentation facilities, which are the base for generating system and WEB documentation, but also online help systems.

2 Running Terminus

You may run Terminus for an existing or a new database.

Requirements

You may run Terminus when you have installed ODABA. Terminus is platform independent and can be used on Windows, LINUX and several UNIX platforms. Terminus requires a project resource database (dictionary), which contains the concept and terminology model object type definitions.

Usage

Terminus can be called from a command line in DOS or UNIX.

```
.../odaba/ode90.exe ini_file [ options ]
```

ode90.exe is the generic GUI driver, which is called as entry point for all ODABA applications. In order to call the proper application, the PROJECT option has to be set to Terminus., which is possible in the configuration or ini-file or as additional option passed on the command line.

Ini-file

The ini_file refers to the database location and can be provided as configuration (xml) or ini-file. Below you will find examples for an ini-file. The DATABASE option in the ODE90 section refers to the database location. In order to create a new terminology database, just replace the location by the path to the new database. When creating the database, you are requested to setup some options for the dictionary.

```
[SYSTEM]
DICTIONARY=ode.sys

[ode90]
PROJECT=Terminus
SYSDB=ode.sys
RESDB=ode.dev
DATDB=projects\TM.dev

NET=YES
SYSAPPL=YES
ONLINE_VERSION=YES

PROJECT_DLL=Designer
CTXI_DLL=AdkCtxi

DESIGNER_RES=res
```

Options

In order to reduce the number of required ini-files and command files, you may use the ini-file as being provided with the installation, which refers to a symbolic database path referenced by the DBPath option. Calling Terminus any number of options can be passed preceded with a '-' sign:

```
ode90.exe cmd/ode.ini -PROJECT=Terminus -DBPath=projects\opa\TM.dev  
-TRACE=log
```

Using options allows running many ODABA with only one ini-file. Instead of passing option parameters, you may also set options in a command or bat file as in the example below. In contrast to options set in command or ini-files, which must never be quoted, options passed on the command line have to be quoted when containing spaces.

DBPath is a option, which has been defined in the ini-file for the DATABASE option in order to support dynamic path locations. Since DATABASE might be referenced in different sections of the ini-file, it is more save to refer to the more specific variable DBPath.

Details for configuring data source are described in o the WEB documentation under Reference documentation/Database references/Data source definition.

```
// ode.cmd for starting Terminus  
set PROJECT=Terminus  
EXPORT Project  
set DBPath=projects\opa\opa.dev  
EXPORT DBPath  
set TRACE=log  
EXPORT TRACE  
START ODE90.exe cmd/ode.ini
```

Samples

There is one Terminus Sample with the Terminus database, which contains the terminology model for the terminology model. It also includes hierarchical topics for this documentation.

3 Overview

Terminus supports defining expert knowledge on different levels. In order to get precise information about expert knowledge, the structure of the terminology model helps to store the right information on the right place.

Usually, the first step is defining relevant terms used in the subject area. The next step will turn simplified concept definitions into terminology model object types or global features (e.g. processes). Getting the details for terminology model object types is the next step. Here, experts are requested to define properties for the different object types. Names for properties are defined by the experts, who know best the proper names for such details.

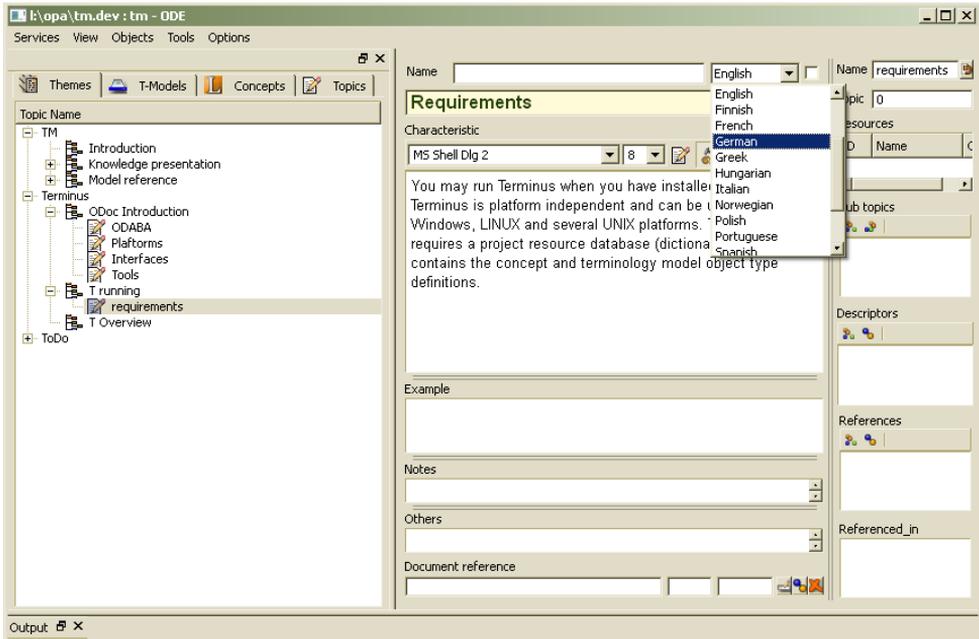
Usually, it is a good idea, to separate property definitions from the more advanced behavior or reaction definitions. Sometimes, however, rules or reactions are discussed when defining properties, and might be defined in this case as well. Often, property definitions result in new object types, which require property definitions again. Moreover, generalizations defined in the concept system may change and become more specific. Thus, the concept system should be maintained in parallel or has to be reviewed later in order to obtain the precise terminology definition for the concepts referenced.

In order to describe complex processes or contexts of the subject area, but also in order to define user's guides like this document, the themes hierarchy can be used for creating structured documents.

Terminus supports generating different sorts of output from the terminology model, concept system and themes list. Open Office or Microsoft word documents can be created as well as WEB documentation in form of HTML sites (as the online documentation for Terminus).

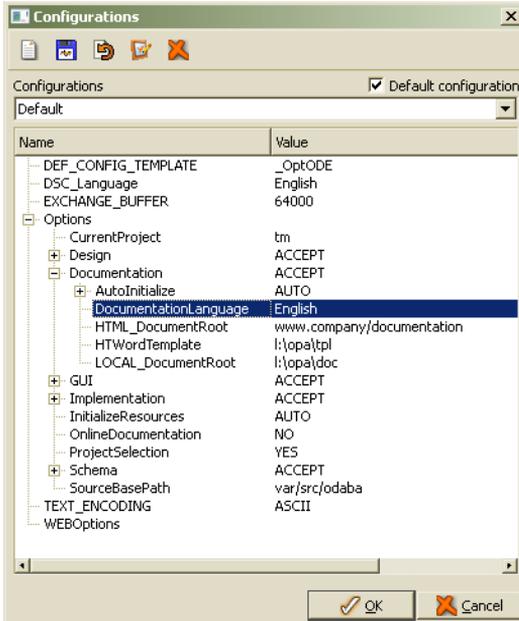
Multi-lingual support

Terminus supports several languages, which might be selected from a language selection box. This allows entering text in different languages, which is required in case of providing multi-lingual applications.



All topics and concept definition text blocks provide the language selection box, which selects the language for all text parts in the topic (characteristic, example, notes and others). Links to other topics or concept relations (generalization, descriptors, references) are not language dependent but might be displayed with language specific texts.

When working with multi-lingual documentation, the developer is responsible for updating additional languages in case of changes in the documentation. There is automatic indication, that other language versions have to be updated.



Usually, the default language is selected automatically, when creating a new topic or concept. The default setting might be changed in the options dialog, which can be opened via the application menu **Options/Setting**.

For new applications, the language might be set also in the ini-file in **Options.Documentation.DocumentationLanguage**.

Changing the language settings for an existing database may result in empty forms, which will be provided for the new language.

When generating documents or WEB-sites from the Terminus database content, the language has to be set to the language for which the documents are to be created before

starting the document generation. When generating documents from OSI or OS-shell scripts, the proper language has to be set in the script or system environment by setting the full option path as described above.

Tools

Terminus provides several tools for generating word or html documents. In order to create a document output for a part of the hierarchy, one of the following options might be selected.

Similar tools are provided in the terminology model tree, where you may create documentation for a complete terminology model, a single object type or a class. Documentation can be provided in general as MS Office or Open Office document.

Terminus also provides translation support. In order to create full text indices for the document topics, index services are provided as well as search function (full text search).

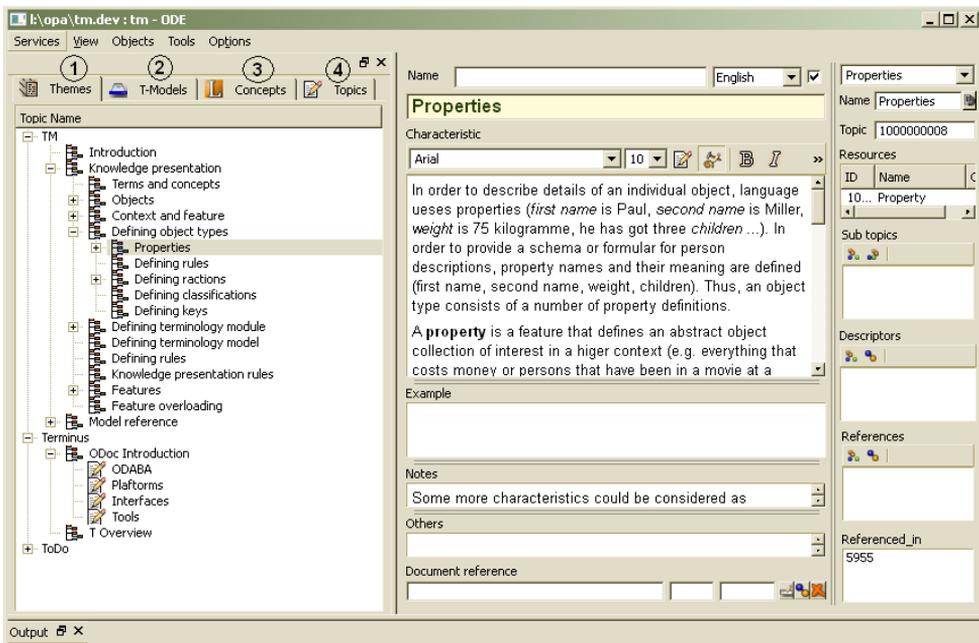
Several tools are provided via the Tool menu in the Terminus main menu. Other tools are available as command line tools.



4 Developing a terminology model

Developing a terminology model may directly start with the terminology model definition. It is suggested, however, to begin with defining most relevant concepts, i.e. defining the meaning of the terms as being used by subject area experts. Often, defining the terminology model is a mix between defining concepts and terminology model object types. In any case, each terminology model object should get a short concept definition in order to provide a fast overview.

Terminus supports the different definition areas of a terminology model by using different views to the terminology model database:



- (1) Provides a view to thematic documentation
- (2) Provides the terminology model view for defining models and object types
- (3) Provides the concept system view for defining concepts and concept relations
- (4) Provides a topic view for experienced users

-

5 Terminus Tools

Terminus tools are provided a built-in application features, but also as external tools, which might be called from a command line or from within the OShell.

Several tools require specific parameters, which have to be set properly in the Terminus options dialog or in the environment calling the feature. Required options are described separately for each tool.

5.1 Global Terminus tools

Global tools are those, which are called from the Terminus main menu bar (Tools). This version supports index services and topic translation. In addition, a search facility is provided in order to search topics by search words.

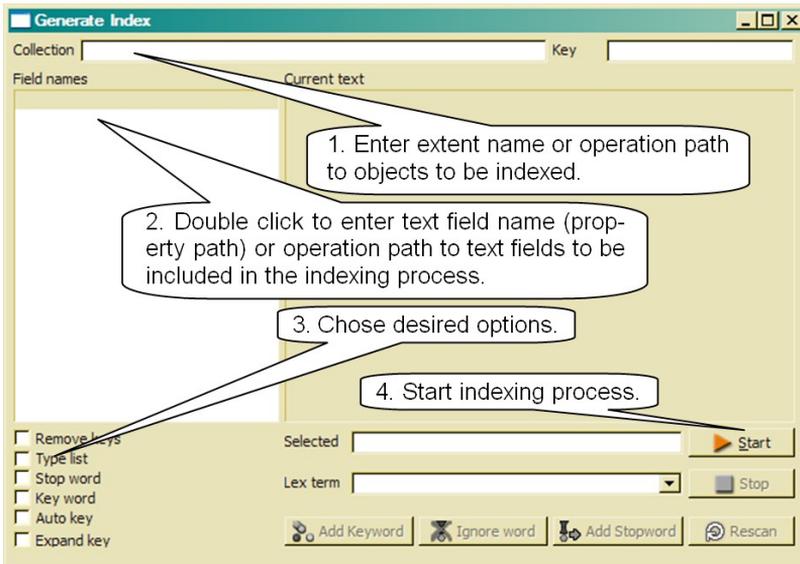
5.1.1 Translation support

Translation support simply provides a two pane window, which allows displaying and editing the text in two languages.

In order to translate the currently selected topic, Translation might be called from the Tool menu in the Main menu bar.

5.1.2 Index services

Index services are provided in order to create a full text search index for topics. In order to update the current index, Update index might be called from the Tool menu in the Main menu bar.



After selecting the menu item, the index service dialog pops up which allows defining the indexing request. In principle, any type of object might be indexed, but Terminus especially supports indexing topics, concepts notices and themes.

Index services can be configured in the index service dialog. More details about running index services are described in Index Services documentation.

In order to avoid re-configuring the index service dialog always when being called, the index manager can be configured in the ini-file by inserting an index manager section (see example below), or in the options dialog by adding index manager options.

Index services might be called from outside, as well. Since indexing may take longer time, it makes sense to index the database explicitly by calling the index manager services.

While indexing, keywords can be defined or text might be corrected. Thus, indexing is also a quality check for the documented topics.

For more information about running the index manager see

<http://www.run-software.com/inhalt/downloads/documentation/IndexServices.doc>

In order to maintain the stop word and keyword list, you may select the Concepts tab (above the list) and then the down tab **Keywords**. or **Stopwords**. A list with all keywords (or stop words) will be displayed and you may remove invalid key or stop words from the list.

```
; section to be added in the Terminus ini-file
[IndexManager]
keywords=DSC_Keyword
stopwords=DSC_Stopword
lexterms=DSC_LexTerm

collection=DSC_Concept.OrderBy(sk_Name)
field1=definition.name
field2=definition.definition.characteristic

stop_word=YES
remove_keys=NO
type_list=YES
```

5.1.3 Text search

In order to find topics by keywords, a search feature is provided on the **Concepts** in the tree pane. After selecting **Concepts** in the upper tab bar, **Search** can be selected in the tab bar below the concept list.



From the type list you may select the whether you want to search for topics, concepts, themes or notices. The default settings are configured for searching topics, but in your application, you may index any other object type.

Another possibility is going directly to the keyword list (tab **Concepts** and down tab **Keywords**), where you may easily select all topics related with a given keyword.