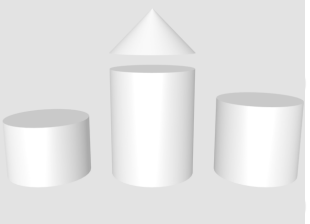


01101001001110010101  
10101101010010111011  
10001011101010101011  
10110010100101011010  
10101001101011010010  
01110010101101011010  
10010111011100010111



**ODABA<sup>NG</sup>**

01110010101101011010  
10010111011100010111  
01010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101  
01010100110011010010  
01110010101101011010  
10010111011100010111  
01010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101

## Documentation Overview

01110010101101011010  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101  
01010100110011010010  
01110010101101011010  
10010111011100010111  
01010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
1101100101001101101  
01010100110011010010  
01110010101101011010  
10010111011100010111  
01010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
1101100101001101101  
01010100110011010010

run

0101011011100010111  
010101011101100101  
001010110101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101  
01010100110011010010  
01110010101101011010  
10010111011100010111  
010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101  
01010100110011010010  
01110010101101011010  
10010111011100010111  
010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101  
01010100110011010010  
01110010101101011010  
10010111011100010111  
010101011101100101  
00101011010101010011  
00110100100111001010  
11010110101001011101  
11000101110101010101  
11011001010010101101  
01010100110



**run Software-Werkstatt GmbH**  
**Köpenicker Strasse 325**  
**12555 Berlin**  
www.run-software.com  
Tel: +49 (30) 65762791  
e-mail: run@run-software.com

Berlin, August 2010

# Content

<b>List of Documents (by areas)</b> .....	<b>4</b>
Concepts.....	4
Real Objects.....	4
Database Concepts.....	4
Glossary and Terms.....	4
Database features.....	4
Replication server.....	5
Workspaces.....	5
Active Data Link.....	5
Multiple database storage support.....	5
Installation.....	5
Installation.....	5
License.....	6
Database Services.....	6
Database Utilities.....	6
Data Exchange.....	6
Document Exchange.....	6
Remote processing.....	6
Index Services.....	6
Object Commander.....	7
Development.....	8
Development tools.....	8
Development features.....	8
Papers.....	11

## List of Documents (by areas)

Documents are assigned to different areas to make it easier distinguishing between conceptual and technical oriented documents.

- Concepts
- installation
- Services\Utilities
- Development
- Papers

## Concepts

Conceptual documents describe ODABA concepts from different perspectives. Some of the documents describe technical concepts, other general database modelling concepts.

### Real Objects

Basic concepts of object-oriented databases in general and ODABA2 in particular have been published in "Real Objects", Addison Wesley 1996 (German). This is an electronic version of the book.

→ [Concepts\RealObjectsGE.doc](#)

### Database Concepts

Database Concepts is an English variant of "Real Objects", which describes the specific features of ODABA. This has been the base for the English version of the ODABA2 concept documentation, as well.

(not yet available)

→ Concepts\ODABAConcepts.doc

### Glossary and Terms

Within ODABA general terms used in object-oriented programming and databases become more specific. The specific meaning of terms is defined in this document.

→ Concepts\ODABAGlossary.doc

### Database features

Database features describe common and special features of ODABA.

→ [Concepts\TechnicalOverview.doc](#)

- Replication server** Replication database in ODABA is a feature, that not only allows creating a database replicate for security reasons, but also supports client in slow network environments as the internet.  
→ [Concepts\ReplicationDatabase.doc](#)
- Workspaces** Workspaces can be used to store updates for metadata objects temporary until updating the metadata object has been finished. The administrator must enable the workspace feature explicitly for each database that should be used with workspaces.  
→ [Concepts\Workspace.doc](#)
- Active Data Link** The paper describes the requirements for a database, that should be run as active database in an ADL environment.  
→ [Concepts\ActiveDataLink.doc](#)
- Multiple database storage support** ODABA supports several database storage types in order to store ODABA data in several relational databases (RDBMS). The document describes how to provide access packages for relational databases other than ORACLE, MySQL or MS SQL Server.  
→ [Concepts\ORMappingConcept.doc](#)

## Installation

- Installation** The Installation Guide describes the main steps for installing ODABA2 and tools. This document is shipped with the installation as well.  
→ [Installation\QuickStarterGuide.doc](#)

**License** License regulations contain the agreements for using ODABA. Practically, you may use ODABA as long as you are not going to sell products based on ODABA. In this case you must buy ODABA developer licenses, for the developers involved in developing commercial products.

→ Installation\License.doc

## Database Services

**Database Utilities** This document contains instructions for running several database utilities and server commands. This includes services as database re-organization, compression or consistency checks.

→ [Utilities\DatabaseUtilities.doc](#)

**Data Exchange** ODABA supports different ways for exchanging data with external sources (import/export) The document describes how to provide data exchange specifications for different data type formats and how to run data exchange.

→ [Utilities\DataExchange.doc](#)

**Document Exchange** ODABA supports different ways for importing and exporting documents (MS Office, Open Office). The document describes different techniques used for document exchange.

→ [Utilities\DocumentExchange.doc](#)

**Remote processing** In order to support distributed processing, ODABA provides remote processing features. The document describes how to use remote processing and how to run remote processing in the ODE.

→ [Utilities\RemoteProcessing.doc](#)

**Index Services** Index Services are used for associating application object instances with a key word index. ODABA<sup>NG</sup> provides index services in order to generate keyword indexes for documentation elements (topics, concepts, themes, notices).

In addition, Index Services support indexing any type of application object. The only requirement is, the application objects to be indexed inherit from the documentation object base (DSC\_Object). Since DSC\_Object is a system type, any application type may inherit from this type just by defining it as base type for the user-defined

type.

→ [Utilities\IndexServices.doc](#)

**Object  
Commander**

Instead of running database utilities, you may use the object commander, which allows browsing to databases, running OShell commands or database utilities.

→ [Utilities\ObjectCommander.doc](#)

## Development

	<p>This folder contains documents describing development tools as well as development features.</p>
<b>Development tools</b>	<p>Development tools provide all features required for building ODABA applications and database definitions. Development tools are available as GUI tools and as command line tools.</p>
Terminus	<p>The document composer is a GUI tool, which allows completing topic and concept documentations, but also combining topics to complex hierarchical document structures.</p> <p>→ <a href="#">Development\Terminus.doc</a></p>
Class Editor	<p>The Class editor provides all the necessary support for defining the data model and the functional model for an application. The current version of the Class Editor supports Program languages C, C++ and OSI.</p> <p>It provides also features for checking data model definitions and compiling and linking implementation classes.</p> <p>→ <a href="#">Development\ClassEditor.doc</a></p>
Designer	<p>The designer is a GUI tool, which allows designing GUI applications and documents based on the data model defined for the project. You can build properly working prototypes without any line of programming. This includes also complex view as tree views to data hierarchies.</p> <p>→ <a href="#">Development\Designer.doc</a></p>
<b>Development features</b>	<p>Development features include script languages and application program interfaces (API). ODABA provides an exhaustive API, but also a JAVA like script interface.</p>
Function Reference	<p>The function reference describes the access classes supported by the ODABA API. This includes data and metadata access as well as accessing run-time information.</p> <p>→ <a href="#">Development\ODABARefrence.doc</a></p> <p>The documentation describes the C++ interface, but most of the functions of the access classes can be called via</p>

the COM interface from .NET, PHP and other program languages.

The documentation is available also as online documentation at (**Reference documentation/ODABA application interface**)

<http://www.run-software.com/content/documentation/odaba>

#### User's Guide

The user's guide describes the way to access data in an ODABA database from within a process (application). It contains a number of examples how to open a database and how to access data in a database.

→ [Development\ODABAUsersGuide.doc](#)

The document has not been finished, but it contains a number of useful hints. Hence we decided to deliver the incomplete version of the document as well.

The documentation is available also as online documentation at (**Users Guide**)

<http://www.run-software.com/content/documentation/odaba>

#### Designer's Guide

The designer's guide describes the way to design an application. It explains the essential steps as creating project window or dialogs, describes how to use design patterns and templates and explains the way of controlling an application via event handlers implemented in context classes.

→ [Development\ODABADesignGuide.doc](#)

The document has not been finished, but it contains a number of useful hints. Hence we decided to deliver the incomplete version of the document as well.

The documentation is available also as online documentation at (**Designers Guide**)

<http://www.run-software.com/content/documentation/odabagui>

#### Designer Reference

The designer's reference guide describes design objects and its properties. It contains a detailed description of the way, design objects behave under certain circumstances.

→ [Development\ODABADesignObjectReference.doc](#)

The document has not been finished, but it contains a number of useful hints. Hence we decided to deliver the incomplete version of the document as well.

The documentation is available also as online

documentation at (**Reference documentation/Design objects**)

<http://www.run-software.com/content/documentation/odabagui>

Class Reference The designer's class reference provides an overview about available GUI context and resource classes and its features.

→ [Development\ODABADesignClassReference.doc](#)

The document has not been finished, but it contains a number of useful hints. Hence we decided to deliver the incomplete version of the document as well.

The documentation is available also as online documentation at (**Reference documentation/Design classes**)

<http://www.run-software.com/content/documentation/odabagui>

Script Interface The ODABA Script Interface (OSI) is a JAVA like programming language. You may write simple queries using OSI but also complex programs or a mixture of it.

→ [Development\OSI.doc](#)

Service classes ODABA2 is referring to a number of service classes, which we started to document. At the moment, the documentation is at the beginning, but it will be improved. Service classes is something you do not really need, but which might be useful in some cases.

→ [Development\ServiceClasses.doc](#)

## Papers

- This folder contains a collection of published and unpublished papers, which provide a little insight into the scientific background of ODABA.
- Terminology Model version1 A terminology model is a mean for recording and presenting expert knowledge in a certain area. The terminology model provides both, a method for discovering subject expert knowledge and a structured and formalized way of presenting expert knowledge. The terminology Model is a method for defining problem items, but also for designing the object model. In this paper it is also shown, how the terminology model has been used as method for defining common statistical expert knowledge from several national statistical offices.  
The terminology model as being described here has been developed in cooperation with experts from several national statistical office. It has been published at the Open Forum Conference in Berlin, 2005.  
→ [Papers\TerminologyModel\\_v1.doc](#)
- Terminology Model version2 The version 2 of the terminology model is an improved version developed by RUN.  
→ [Concepts\TerminologyModel\\_v2.doc](#)
- Unified Database Theory The paper provides a common mathematical database theory that unifies relational databases, data warehouse models and object-oriented database models. It is based on long experiences reflected in human language models and database system development. This is a short version of the theory. The detailed version is not yet complete.  
→ [Papers\UnifiedDatabaseTheory.doc](#)
- DBMS Comparison This paper compares the ODABA database management system with other OODBMS.  
→ [Papers\DBMSComparison.doc](#)
- RDBMS and OODBMS This paper compares the ODABA database management system with other OODBMS.  
→ [Papers\RDBMSandOODBMS.doc](#)
- Efficient Modelling Techniques This paper demonstrates theoretical and practical approaches for designing and implementing complex

data models. It is based on research work made for developing the OODBMS ODABA2 as well as on practical experiences. Two medium object models will be used as examples: the Integrated Metadata System for statistical offices **Bridge**, developed within the frame of the DOSIS project (EUROSTAT) and an administration system for large enterprises **Belami**. Both consist of more than 100 persistent object types (including more than 300 relationships) and about 300 additional transient classes (implementation classes). Those correspond to a relational database model with 500-1000 physical tables.

→ [Papers\ModellingTechniques.doc](#)