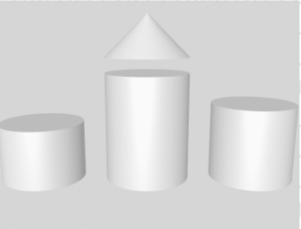

ODABA^{NG}

Documentation Overview

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

List of Documents (by areas)

Documentation Overview

Documents are assigned to different areas to make it easier distinguishing between conceptual and technical oriented documents. Document categories are coded in the initial character of the document name.

- Introductions (0)
- Concepts (1)
- Installation (2)
- Services\Utilities (3)
- Development (4)
- Papers (P)

→ 0.1_DocumentationOverview.pdf (this document)

Introductions

The document provides a short overview about RUN-Software, its activities, products and partnerships.

→ 0.2_AboutRun.pdf

ODABA Invitation

The intention of the paper is to introduce inexperienced readers into the world of ODABA, its technical features, tools and development technology.

→ 0.3_ODABAInvitation_EN.pdf

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 the electronic version of the book.

→ 1.1_RealObjectsGE.pdf

Object-oriented views

In contrast to relational views, object-oriented views are more powerful and simpler to handle. By means of examples the paper introduces the idea of view or access paths. .

→ P5_OOViews.pdf

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.

(unfinished)

→ 1.2_ODABAConcepts.pdf

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.

→ 1.3_ODABAGlossary.doc

Database features

Database features describe common and special features of ODABA.

→ 1.4_TechnicalOverview.pdf

Replication Database

Replication database in ODABA is a feature, that not only allows creating a database replicate for safety reasons, but also allows running databases on a WEB server. Thus, clients distributed all over the world may access the same ODABA database.

→ 1.5_ReplicationDatabase.pdf

Workspaces

Workspaces are a special ODABA feature, which provide persistent transactions. Workspace transactions will survive the process and are available for any number of processes. Results of a workspace transaction are stored in a separate database and are copied to the original database when being consolidated by the user.

The workspace features can be enabled running the Workspace database tool.

→ 1.6_Workspace.pdf

Active Data Link

The paper describes the requirements for a database, that should be run as active database in an ADL environment.

→ 1.7_ActiveDataLink.pdf

OR Mapping

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 as ORACLE, MySQL or MS SQL Server.

→ 1.8_ORMappingConcept.pdf

Installation

Quick Starter's Guide

The document describes the main steps for installing ODABA and tools. This document is shipped with the installation as well.

→ 2.0_QuickStarterGuide.pdf

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. Among others, this includes services as database re-organization, compression or consistency checks.

→ 3.1_DatabaseUtilities.pdf

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.

→ 3.2_DataExchange.pdf

Document Exchange

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

→ 3.3_DocumentExchange.pdf

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.

→ 3.4_RemoteProcessing.pdf

Index Services

Index Services are used for associating application object instances with a key word index. ODABA^{NG} 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.

→ 3.5_IndexServices.pdf

Object Commander

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

→ 3.6_ObjectCommander.pdf

Development

This folder contains documents describing development tools as well as development features.

Development tools

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.

Terminus

Terminus is a GUI tool, which allows defining terminology models, completing topic and concept documentations, but also combining topics to complex hierarchical document structures. Terminus provides several tools as indexing or model transformation.

→ 4.1_Terminus_(unfinished).pdf

Class Editor

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.

It provides also features for checking data model definitions and compiling and linking implementation classes.

→ 4.2_ClassEditor_(unfinished).pdf

Designer

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.
→ 4.3_Designer_(unfinished).pdf

Development features

Development features include script languages and application program interfaces (API). ODABA provides a exhaustive API, but also a JAVA like script interface.

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.

→ 4.4_OSI.pdf

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.

→ 4.5_ODABAUsersGuide.pdf

ODABA Class Reference

The ODABA class reference describes the access classes supported by the ODABA API. This includes data and metadata access as well as accessing run-time information. Functions are described according C++ conventions, but are available in C# and OSI, too.

□ 4.6_ODABAReference.pdf

GUI Class Reference

The GUI class reference provides an overview about available GUI context classes and resource classes and its features.

→ 4.7_GUIReference.pdf

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.

→ 4.8_DesignReference.odt

Papers

This folder contains a collection of published and unpublished papers, which provide a little insight into the scientific background of ODABA.

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.

→ P1_UnifiedDatabaseTheory.pdf

Database Categories

The paper illustrates the differences between main database categories and shows the difference between simple and intelligent databases.

→ P1_DatabaseCategories.pdf

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.

→ P2_TerminologyModel_v1.pdf

Terminology Model version2

The version 2 of the terminology model is an improved version developed by RUN.

→ P2_TerminologyModel_v2.pdf

DBMS Comparison

This paper compares the ODABA database management system with other OODBMS.

→ P4_DBMSComparison.pdf

RDBMS and OODBMS

This paper compares the object-oriented database management system with relational DBMS.

→ P3_RDBMSandOODBMS.pdf

Efficient Modeling 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.

→ P6_EfficientModeling.pdf