ODABA Releases TODBMS and Tools 17.2.1

ODABA is a Terminology-Oriented Database Management System (TODBMS) based on standards for object-oriented databases (ODMG 2003). In contrast to other databases that are focused on big data processing, ODABA stands for smart data processing, i.e. it is intended to be used for complex problems and complex data structures in combination with complex processing rules.

The latest version of ODABA has been released on Sunday, August 29th, 2021. With ODABA 17.2.1 a version with several bug-fixes and slight changes has been provided. For Windows users, a DevStudio 2017 compiled version is released in addition for 32 and 64 bit. For Linux users, GCC 6 is supported.

We do not deliver anymore .msi files for Windows installations, since ODABA does not need any kind of registration in the Windows registry. Instead, binary installations for Windows are delivered as 32 and 64 bit versions compiled with MS VS2010 and VS2017.

More details are described in change logs and in notices delivered with the development databases (ODE tools: **Objects/Notices**). Notices delivered with the databases also contain a list of open topics planned for next releases. Notices are stored separately for basic functions (**sos.dev**), database kernel (**opa.dev**), GUI framework (**gui.dev**) and ODE tools (**ode.dev**).

run Software-Werkstatt GmbH
Winckelmannstrasse 61
12487 Berlin

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

Berlin, August 2021

Detailed changes (ODABA)

The behavior of several features have been improved. Several bugs have been removed. Removed bugs are reported in the change log. Also, several interface functions fave been changed.

ODABA Database kernel (base)

Several ODABA components have been improved or provide extended features:

Database features

- Transient property handle
  When opening/setting a transient property handle, the referenced property handle will be opened and type compatibility is checked.

- FILE_CATALOG and DATA_CATALOG keyword
  Previous versions (FILE-CATALOGUE, DATA-CATALOGUE, FILE-CATALOG and DATA-CATALOG) are not supported anymore, i.e. client/server configuration files have to be updated.

- Convert file paths to system specific format
  Before calling system functions, passed file names are converted into system specific syntax ('\' or '/')

- Ignore RDB state
  Calling utilities locally, will reset the replication database state, which is not necessary in many cases. In order to access replication databases for maintenance purposes, the RDB state of the database may be ignored by setting the option IGNORE_RDB.

Utility parameters

Several utilities support symbolic names, which may include explicit file paths, option names included in %...% or references to configuration file. Thus, it becomes possible, running several utilities in client/server mode, also. Utility parameters have been changed for following services:

- BackupDB (symbolic name instead of ini-file)
- CheckDB (additional optional parameters)
    - CopyDB (additional optional parameters)
    - DBDictStatistics (symbolic name instead of dictionary path)
    - DBReplication (symbolic name instead of database path and keyword parameters)
- DBSchemaVersion (symbolic name instead of database path)
- DBStatistics (symbolic name instead of database path)
- DBSystemInfo (symbolic name instead of database path)
- DBSystemVersion (symbolic name instead of dictionary path)
- DBVersion (symbolic name instead of database path)
- PackDB (symbolic name instead of database path)
- ResetKeyLocks (symbolic name instead of database path)
- RestoreDB (symbolic name instead of ini-file)
- SetupDB (symbolic name instead of ini-file path)

Event handling:.

- Do not call event function when related object is destroying
  Since the state of the object going to be destroyed if undefined, calling events in this state may cause problems.

- Close client when server sends a `SE_Close` event

- Generate Delete event when instance is going to be deleted

ODABA Application Program Interface (base/opa)

Several extensions and some changes have been made on API functionality. Especially, the return value of all math functions in `Value` class changed from `Value` to `double`.

Most interface classes inherit now from ObjectReferenceHandle, which supports multiple use of referenced data objects. Hence, most interface classes got an assign function and/or an operator=.

Change status has following meaning:

- new - New function, class, enumeration or enumerator

- updated - Function has been updated
- expanded - Functions with same name but different parameter lists have been added
- removed - Function has been removed from interface
- return - return value data type changed
- osi - Function has been added to OSI interface

Interface changes:

- Property
  - closeAll (expanded, osi)
  - moveDown (return)
  - moveUp (return)

- **Value**
  - abs (return)
  - acos (return)
  - asin (return)
  - atan (return)
  - cos (return)
  - cosh (return)
  - exp (return)
  - log (return)
  - log10 (return)
  - pow (return)
  - sin (return)
  - sinh (return)
  - tan (return)
  - tanh (return)
  - sqrt (return)

- **DataSource**
  - assign (new, osi)

- **Database**
  - assign (new, osi)
  - closeMirror (new, osi)
  - enableReplication (expanded, osi)
  - operator= (new)

- **BaseContext**
  - message (expanded, osi)
  - parentContext (renamed from 'parent')

- **ObjectSpace**
  - assign (new, osi)
  - operator= (new)

- **DateTime**
  - operator= (new)

- **PropertyDefinition**
  - assign (new, osi)
  - operator= (new)

- **Client**
  - assign (new, osi)
  - receiveFile (expanded, osi)

- **TypeDefinition**
  - operator= (new)

- **Binary**
  - Binary (expanded, osi)

- **TypedString**
  - assign (expanded, osi)
  - compare (new, osi)
  - copy (new, osi)
  - data (new)
  - operator!= (new)

- operator+ (new)
- operator+= (new)
- operator< (new)
- operator<= (new)
- operator== (new)
- operator> (new)
- operator>= (new)
- toString (new, osi)

- **Dictionary**
  - assign (new, osi)
  - operator= (new)

- **Application**
  - workDictionary (expanded, osi)

- **Time**
  - operator= (new)

- **Date**
  - operator= (new)

- **Key**
  - data (moved to **TypedString**)
  - isEmpty (updated)
  - operator!= (moved to **TypedString**)
  - operator== (moved to **TypedString**)
  - toString (moved to **TypedString**)

- **StringList**
  - operator= (new)

- **EnumeratorDefinition**
  - assign (new, osi)

- **BNFNode**
  - assign (new, osi)
  - operator= (new)

- **BNFParser**
  - assign (new, osi)
  - operator= (new)

- **GlobalVariable**
  - assign (new, osi)

- **IndexDefinition**
  - assign (new, osi)
  - operator= (new)

- **Email**
  - assign (new, osi)
  - operator= (new)

- **HTTP**
  - assign (new, osi)
  - operator= (new)

- **File**
  - File (expanded, osi)
  - operator= (new)

- **TextFile**
  - read (expanded, osi)
  - operator= (new)

- **INIFile**
  - operator= (new)

More details are described in ODABA online documentation: **Reference documentation/ODABA Application Program Interface.**

ODABA Script Interface OSI

OSI interfaces have been provided for all new (or changed) interface functions. Checking OSI functions has been improved by checking parameter lists and parameter data types.

A new style of generating and executing OSI-interface functions has been provided in order to support checking parameter types. All interface functions (`_osi_fctname()`) have been rewritten.

Other changes are:

- `binary` keyword changed to `bin` (in order to avoid conflicts with **`Pixmap::`**`binary()` function)
- When displaying array attributes in OShell or OSI debugger (p or print command), now all array elements are displayed

New function / operations supported in OSI are:

- `<<` operator for text fields
- `>>` operator for text fields

Open Document Support

Open document support has been improved and some bugs have been removed.

Detailed changes (ODE and GUI framework)

Some bugs in managing GUI resources have been removed. Support for formatted data input has been provided and some extensions have been made in ODE tools.

GUI Framework (gui)

Some bugs in managing GUI resources have been removed. Support for user defined wait pointers has provided. Some extensions have been made in ODE tools.

ODE tools (ode)

Some improvements have been made in different tools and components.

Action-Log

- Explicit function for terminating recording the session

ODABA GUI Application Program Interface (gui/ode)

Several extensions and some changes have been made on API functionality. Most interface classes inherit now from ObjectReferenceHandle, which supports multiple use of referenced data objects. Hence, most interface classes got an assign function and/or an operator=.

Change status has following meaning:

- new - New function, class, enumeration or enumerator
- updated - Function has been updated
- expanded - Functions with same name but different parameter lists have been added
- removed - Function has been removed from interface
- return - return value data type changed
- osi - Function has been added to OSI interface

Context classes

- **ActionItem**
  - copy (new, osi)

- **Line**
  - copy (new, osi)
  - line (updated)

- **Color**
  - assign (new, osi)

- **Column**
  - copy (new, osi)

- **ControlContext**
  - changeOrderColumn (expanded, osi)
  - expandToItem (expanded, osi)
  - setDisplayLevel (expanded, osi)

- **Font**
  - assign (new, osi)
  - operator= (new)

- **GUIBaseContext**
  - deactivateActionLog (new, osi)
  - executeAction (expanded, osi)
  - resetWaitPointer (new, osi)
  - setWaitPointer (new, osi)
  - setWaitPointerGeneric (new, osi)

- **Icon**
  - operator= (new)

- **Image**
  - operator= (new)

- **KeyInput**
  - assign (new, osi)
  - operator= (new)

- **Line**
  - copy (new, osi)

- **Pixmap**
  - operator= (new)

- **Size**
  - assign (new, osi)
  - operator= (new)

Resource classes

No changes made for resource classes.

ODABA Documentation

The documentation tree has been extended by adding new function descriptions and extended utility documentation.

Installing ODABA

ODABA, including applications and libraries, is available for free under Open Source licenses (GPL). ODABA runs on various hardware configurations, operating systems and works on many desktop environments. ODABA can be obtained as source code distribution and in various binary formats from http://sourceforge.net/downloads/odaba/.

Several features require third party components, which have to be installed before installing ODABA. When the corresponding libraries are not available, one may install ODABA, but the features referenced below will not work.

- libzip - required for LibreOffice document generation
- zlib - required for data compression and database backup and restore)
- curl - required for enhanced email support)
- hunspell - required for spell check in ODE tools, like terminus
- libmicrohttp - required for OHTTPServer(D)
- Qt4 or Qt5 - for running the ODABA GUI framework

Using optimizing compiler GCC 6, `this` pointer checks must not be optimized. Use `-fno-delete-null-pointer-checks` option when using GCC optimizing compiler.

Previous Releases

With the release of ODABA 17.2.1 we declare the end of live for all previous released ODABA versions less than version 17.1.0. Bug fixes on 16.x.x version are provided on demand.

System model has not been changed and no version upgrade is required.

System Requirements

In order to get the most out of this release, we recommend to use a recent computer with at least 1 GB of memory and 2 GHz CPU or better. In order to install the binaries, about 250 MB are required. Installing sources requires about 300 MB. 120 MB are required in addition, when installing the documentation locally.

About RUN-Software

RUN-Software develops database management system ODABA and tools since 1994. Besides general and particular software solutions, RUN-Software publishes theoretical works about database theory and terminology in connection with data modeling.

See also: www.run-software.com