

# OpenSCADA 0.8.0 LTS

(Release notes)

## Table of contents

<a href="#">Introduction</a> .....	1
<a href="#">1 Implementation of planned tasks</a> .....	4
<a href="#">2 Optimization, stability, sustainability and system performance increasing</a> .....	5
<a href="#">3 Improvement and stabilization of the graphics subsystem</a> .....	8
<a href="#">4 Formation, expansion and stabilization of the user programming API</a> .....	10
<a href="#">5 System-wide extensions</a> .....	11
<a href="#">6 Publication of OpenSCADA solutions</a> .....	14
<a href="#">Conclusion</a> .....	14
<a href="#">Links</a> .....	14

## Introduction

The release of an open SCADA (Supervisory control and data acquisition) system of version 0.8.0 is the stable industrial release with the long term support (LTS).

The main purpose of this release is providing the community of users and developers of free software with the platform for the solutions of complex automation systems and related solutions, as well as providing commercial services on the basis of the OpenSCADA project.

This release is the next stable release, for which technical support is available from the developers and for which it is planned to release fixes during a long time. The life cycle of the previous stable release 0.7.0 LTS will be terminated within one month after the 0.8.0 release by the latest update.

This document is the processing (compilation) of <ChangeLog> document of the OpenSCADA system version 0.8.0, including the intermediate working releases 0.7.1 and 0.7.2, which is designed to cover brief and clear the new features of the OpenSCADA system. To get acquainted with all the changes in the OpenSCADA system it is possible in the file ChangeLog from the distribution kit of the system, or here (RU): <http://wiki.oscada.org/Works/ChangeLog>.

The key features of this version are:

- Implementation of planned tasks.
- Optimization, increasing of stability, sustainability and system performance.
- Improvement and stabilization of the VCA.
- Expansion of the user programming API.
- System-wide expansions.
- Publication of OpenSCADA solutions.

New and updated modules:

- *Archive.DBArch (0.9.5)* — The limits for requesting data on the request time are added as well as some more extensions.
- *Archive.FSArch (1.5.1)* — Significant stabilization. The set of extensions, including:
  - The files' indexes of the values' archives read performance is increased.
  - The adaptation of storing and reading the floating to the ARM FPA.
  - The limits for requesting data on the request time are added.
  - The limit on the total files' size of the values' archiver is added.
- *DB.MySQL (1.7.1)* — Stabilization. The ability to specify the connection timeouts is added.
- *DB.SQLite (1.6.4)* — Stabilization
- *DB.FireBird (0.9.7)* — Stabilization
- *DB.PostgreSQL (0.9.2)* — Stabilization
- *DAQ.DiamondBoards (1.2.5)* — The ability to change the parameter's type is added. Stabilization.
- *DAQ.System (1.7.5)* — Stabilization. The dependence on the library "Libsensors" is made the optional one. The CRON scheduling of calls is added.
- *DAQ.BlockCalc (1.6.0)* — Stabilization. The set of extensions, including:
  - The syntax highlighting is added.
  - The CRON scheduling of calls is added.
  - The start and stop blocks' call is added.
  - The support of the object parameters' type is added.
- *DAQ.JavaLikeCalc (2.0.0)* — Significant stabilization. The set of extensions, including:
  - The syntax highlighting is added.
  - The adaptation to the ARM hardware architecture is completed.
  - The implementation of regular expressions and a number of user's API extensions are added.
  - The ability to write the string characters using the octal "\041" and the hex "\x21" numbers is added.
  - The direct, dynamic call of library functions is added.
  - The direct association of string constants is added.
  - The conditional call of the expression's arguments ( the second argument) with the logical operators ||(OR) and &&(AND) is added.
  - The support of the data source object parameters' type is added.
- *DAQ.LogicLev (1.3.0)* — Stabilization. The set of extensions, including:
  - The ability to change the parameter's type is added.
  - The CRON scheduling of calls is added.
  - The support of the object parameters' type is added.
- *DAQ.SNMP (0.7.0)* — Significant stabilization. The set of extensions, including:
  - The implementation of the full support of the protocol's client-side is done.
  - The support of all values' types is added.
  - The support of the values' record is added.
  - An error handling is improved.
  - The additional connection configuration parameters are added.
  - The the ability to directly query the scalars is added.
- *DAQ.Siemens (1.4.0)* — Significant stabilization. The set of extensions, including:
  - Library LibnoDave is updated to the 0.8.4.6 version.
  - The CRON scheduling of calls is added.
  - The reconnection function for the "Industrial Ethernet" connections is added.
  - The ISO\_TCP version of the protocol is added for S7-200.

- The support of the object parameters' type is added.
- The "ADS" protocol implementation is added.
- Connection errors are some unified.
- The ability to set the database number and offset in hexadecimal is added.
- *DAQ.ModBus (1.3.0)* — Significant stabilization. The set of extensions, including:
  - The attributes' syntax highlighting is added.
  - The support for "Logical" parameter's type is added for the working with template of the parameter, and also the functions of the user's API to send arbitrary, non-standard ModBus requests from the templates are added.
  - The function of the formation the alarm messages in the controller is added and used.
  - The support of the function for group record (0x0F, 0x10) is added.
  - The option to install the block size limit of the group requests is added.
  - The support of the object parameters' type is added.
  - The maximum size of the request block is made optional.
  - The ability to comment the items in the list of attributes of the standard parameter's type by the '#' at the beginning is added.
- *DAQ.DCON (0.5.1)* — The module is completely rewritten for the support of mixed modules by selecting the request commands. Stabilization.
- *DAQ.ICP\_DAS (0.8.0)* — The "ICP DAS" API library libi8k.a for the ARM architecture is added, the ability to build the module for ARM is added. The build is disabled by default to prevent attempts to build on the wrong ARM environment. Stabilization.
- *DAQ.DAQGate (0.9.5)* — The parameters' cache for their initialization during the autonomous operation is added. The CRON scheduling of calls is added. Stabilization.
- *DAQ.SoundCard (0.6.2)* — Stabilization.
- *DAQ.OPC-UA (0.6.2)* — Stabilization. The syntax highlighting of the attributes' list is added. The adaptation of storing and reading the floating to the ARM FPA is done.
- *DAQ.BFN (0.5.1)* — New module.
- *Protocol.ModBus (0.6.4)* — Stabilization. Some improvements.
- *Protocol.OPC-UA (0.6.2)* — Stabilization.
- *Transport.Sockets (1.5.1)* — Significant stabilization. The outgoing transport request retry timeout is added.
- *Transport.SSL (1.0.1)* — Stabilization. Some improvements.
- *Transport.Serial (0.8.0)* — Stabilization. During initialization of the port the previous settings are used and an opportunity to omit some of the settings is implemented. The flow control using RTS signal for the simple RS232->RS485 signals transducers is added.
- *Protocol.HTTP (1.6.0)* — Significant stabilization. The set of extensions, including:
  - The support for HTTP variable "Transfer-Encoding = chunked" in the outgoing transport is added.
  - The support of user's templates for internal contents of the module is added.
  - The generation of the user's authentication messages is added.
  - The support of all major versions of the completion the string during HTTP-request parsing is added.
- *Protocol.SelfSystem (0.9.5)* — Stabilization.
- *Protocol.UserProtocol (0.6.2)* — The syntax highlighting is added. Stabilization.
- *Special.FLibComplex1 (1.1.0)* — The direct, dynamic call of library functions is added.
- *Special.FLibMath (0.6.0)* — The direct, dynamic call of library functions is added.
- *Special.FLibSYS (1.0.0)* — The direct, dynamic call of library functions is added.
- *Special.SystemTests (1.5.1)* — Some improvements.
- *UI.QTStarter (1.7.0)* — Stabilization. Significant extensions and improvements.
- *UI.QTCfg (2.1.1)* — Stabilization. Significant extensions and improvements.
- *UI.WebCfg (1.5.6)* — Stabilization.
- *UI.WebCfgD (0.8.1)* — Stabilization. Significant extensions and improvements.
- *UI.VCAEngine (1.3.0)* — Stabilization. Significant extensions and improvements.
- *UI.Vision (1.3.0)* — Stabilization. Significant extensions and improvements.
- *UI.WebVision (1.0.1)* — Significant stabilization. Significant extensions and improvements.
- *UI.WebUser (0.6.2)* — Stabilization. The syntax highlighting in the procedures is added.

# 1 Implementation of planned tasks

In accordance with the [plane](#) of the release there were made the following tasks:

- *Formation of the provision of commercial services based on OpenSCADA.* - The concept is built and the mechanisms are created to provide commercial services by the developers based on OpenSCADA (<http://oscada.org/en/services>).
- *Adaptation of OpenSCADA to work on the ARM hardware platform.* - The building, adaptation and full testing of OpenSCADA on the ARM architecture is made on the [N800](#) Internet tablet of the [Nokia](#) company (<http://wiki.oscada.org/Works/Tests/ARM> (RU)). The build and adaptation of OpenSCADA for complex (very old) software environment of the [LP-5451](#) controller are completed, as well as build for the [SMH2Gi](#) controller and [Nokia](#) smart phones: [N900](#), [N950](#), [N9](#) are completed also. As part of the build and adaptation to different mobile devices the following tasks were done:
  - *SYS:*
    - Functions' prototypes `TMess::put()`, `TMess::get()`, `TCntrNode::grpSize()`, `TCntrNode::grpId()`, `TCntrNode::grpAt()` are changed to correct the usage of an unsigned type "char" on the ARM as a signed.
    - The adaptation to build with Glibc version less than 2.5 (2.3.2) is done.
    - *TVariant, DAQ.{AMRDevs, OPC\_UA}*: The not aligned access to data is fixed.
    - *DAQ.OPC\_UA*: The system functions `floatLE()`, `floatLErev()`, `doubleLE()`, `doubleLErev()` are added to convert the storage format of a real number on different architectures.
    - The adaptation to uClibc is done. Build begins with 0.9.32 version, which has the support of real-time functions such as `clock_nanosleep()`.
    - The check and disable the usage the "Iconv" and "Intl" (I18N) API is added.
    - The automatic detection and enabling the `--enable-CrossCompile` configuration option for cross-compilation environment is added.
    - The `--enable-CoreLibStatic` configuration option to disable the OpenSCADA kernel shared library build and static linking of the call-launch program is added.
    - The `usleep()` function is everywhere replaced by the own wrapper function `TSYS::sysSleep()` due to lack of the `usleep()` function in uClibc and availability of it only in the case of special configuration of the uClibc build. The `nanosleep()` is used at the `TSYS::sysSleep()` function.
    - The `pthread_yield()` function is replaced by the `sched_yield()`.
  - *DAQ:*
    - *JavaLikeCalc*: Virtual Machine is rewritten for direct using the structures of commands to avoid alignment problems, as well as for some increasing of productivity.
    - *Siemens*: The problem of the NoDave library for a global packing structures command calling "#pragma pack(1)" is fixed.
    - *ICP\_DAS*: The "ICP DAS" API library `libi8k.a` for the ARM architecture is added, the ability to build the module for ARM is added.
  - *Archive:*
    - Functions' prototypes `TArchiveS::messPut()` and `TArchiveS::messGet()` are changed to correct variable of the message level as a signed number.
    - *FSArch:*
      - The LE format real number storage in the archive on the ARM architecture is fixed.
      - The algorithm of the quickly count the number of bits uses the function of not aligned reading `TSYS::getUnalign32()`.
      - The encoding field size in the plane archive of messages expanded from 9 to 99 characters.
    - *UI.QTCfg*: The short time sleep after closing all windows in the switch off function is added, in order to provide the time for the object of all the windows to be finally destroyed and to prevent crashing.
- *The implementation of the editing changes rollbacks mechanism in the Vision.* — As part of the

widgets' visual editing window the multi-level changes rollback mechanism is implemented for all basic operations: visual geometry changes, the widget attribute's value change, add/remove widgets, widget's copy and edit the widgets, based on the "ElFigure" primitive.

## 2 Optimization, stability, sustainability and system performance increasing

Since the latest stable release During the work on this version, as well as during its practical adaptation, it was discovered and corrected in a total of about 300 errors. In general, the work on cleaning code from the compiler's warnings with the "-Wall" flag has been done:

- unused variables removing;
- adding the return in the all functions;
- initialization of variables with the possibility of their undefined using;
- exception of the incomplete handling of the enumeration types in the select operation;
- comparison between signed with unsigned integer.

Lets list the most significant errors the correction of which significantly affected the increasing of stability:

- *System-wide:*
  - The command of the building system "\$ make dist" is corrected and adapted to correct creation of the OpenSCADA distributions.
  - The *TSYS::cron()* function to call on the schedule is repeatedly corrected.
  - The variable type to store the result of the *find()* function, the string, is everywhere changed to the correct one "size\_t".
  - The unscheduled calls on the schedule, related to desync of the functions values *time()* and *clock\_gettime(CLOCK\_REALTIME,&sp\_tm)*, are fixed.
  - The checking and waiting for threads to access due to the overlapping with the previous, eponymous one's stop are fixed.
  - The waiting for initialization of the disconnectable tasks is disabled in order to prevent freezing in anticipation of rapidly closing tasks.
  - The work of not aligned reading functions *getUnalign\*()* is fixed on the ARM. The Problem is detected on PXA270.
  - *Function:* The possibility of a memory leak when attempting to set the object to a non-object function's parameter is excluded.
  - *DAQ.ModBus, Transport.Serial, UI.{WebCfg, WebCfgD, WebUser, WebVision}*: The incorrect usage of the completion string symbol '\n' is fixed for a variety of software platforms UNIX, MAC, DOS/Windows by replacing it with "\x0A".
  - *ResString:* The "string(vl.data(), vl.size())" way is used for read-write of string in order to prevent the COW algorithm.
  - *Function:* The check for a modification during the installation of the various IO types values (Real to Integer) is fixed.
- *Data bases:*
  - *DB.PostgreSQL:* The handling of database address to determine the empty fields is fixed.
- *Transports and protocols:*
  - *Transport:*
    - The loading of all transports from the database is fixed, if there are records about the transport the module of which is missing.
    - The using of unsigned type in the return of the *read()* and *write()* functions is fixed. The resources for the counters of the i/o requests are added.
    - The option of interrupting the outgoing transport by the signal on the timeout waiting. It is useful to interrupt the transport on exit/shutdown.
  - *Sockets:*
    - The missing of the initialization of the size of the return value of the *getsockopt()* function is fixed.
    - The closing of the client connection by the time out of the lifetime for

- connections in the case of the data absence is fixed.
- Unix-socket is set to nonblocking mode by the `O_NONBLOCK` flag. The generating of the error messages for outgoing transports is fixed.
- *Serial*: `O_NONBLOCK` flag is set to `open()` function of a serial device to prevent the blocking of port's opening on some USB->RS485 converters.
- *Sockets, SSL*: The cleaning of the input buffer to prevent re-use on the high latency channels after timeout is added.
- *Transport, DAQ*.{*AMRDevs, DCON, ICP\_DAS, ModBus, OPC-UA*}, *Protocol.HTTP*: The usage of the `TTransportOut::messIO()` function at the request of the tail and getting the zero response is fixed.
- *Protocol.HTTP, UI*.{*WebCfg, WebCfgD, WebVision, WebUser*}: The parsing of POST-request elements is fixed by the for parsing all the attributes of each element.
- *Data acquisition and archives*:
  - *Archive*:
    - The using of the buffer with a soft grid with low-and high-resolution time in the archives of values is fixed.
    - The calculation of the archive's beginning in the `begin()` function with an empty buffer is corrected.
    - *FSArch*:
      - The creation of messages archive's files during the placing of the set of old messages is fixed. The parameter to prevent duplicates is added.
      - The result of the `fgetc()` function is recorded in an integer variable instead of a character one that has corrected the discovery of "EOF".
      - The reading of data from the archive's buffer in accordance with the size of the buffer in the algorithm of fast calculation of the bits number is fixed.
      - The placement of the offset into the cache from the algorithm of fast calculation of the bits number is fixed.
      - The incorrect processing of the not aligned blocks end of the fast bits number counting algorithm is fixed.
      - The exclusive resource to the `calcVIOff()` function call is added to prevent the incorrect operation with cache.
  - *DAQ*:
    - *System*: The crash when exiting to the "Hddtemp" source is fixed.
    - *ModBus*: The setting of values of the parameter's attributes mode is changed to active and the error handling is improved.
    - *JavaLikeCalc*:
      - The check of the index when accessing the character of the string on the index is added, in order to prevent the out of range situation.
      - The crash of the `replace()` function of the "RegExp" object is fixed.
    - *Siemens*:
      - The receiving and recording the values of real types is fixed.
      - The reconnect in the case of error is fixed.
      - The using of the integer, real and string size property is fixed.
      - The cleaning of the parameter's links at its disabling is added.
- *Visualization Interfaces*:
  - *VCAEngine*:
    - The blocking of the session for dynamically active projects with dynamic access to their own widget's attributes is fixed.
    - The possibility of changes' loss in the visualization server is precluded when displayed in the visualizers by preliminary saving the value of clock counter in the "openList" service request.
    - Resources to widget's attributes is made more clear (by use Recursive mutex) and use one generic resource, for session and other, for each other dead locking prevent.
  - *Vision, WebVision*:
    - The resolution of the coordinates is increased of all the widgets to three decimal places, in order to ensure proper positioning on the large scales.

- The access abroad the vector's size during the rendering of the primitives "FormEls" graphics is fixed.
- The redraw of the primitive "Elementary figure" is fixed when the scale is changed.
- The update of the "Protocol" primitive in time and size is fixed.
- *Vision*:
  - The crash of the session for projects with an execution period of more than 500 ms is fixed.
  - The fine-tuning of the coordinates of the "Elementary figure" primitive shapes is implemented to the remainder of the widget's position.
  - Scaling the text primitive "Text" is changed to change the font size instead of scaling the final image, which increased the overall quality of the display font.
- *WebVision*:
  - The validation of figures' points for their getting to the display area of the primitive "Elementary figure" is removed. This function is shifted to the GD library.
  - The possibility of crashing for primitives "Diagram" or "ElFigure" is excluded in the case of transmission of incorrect values of attributes of the image "xSc" or "ySc" from a Web-browser.
  - The opening and replacing the root page is fixed.
  - The handling of nested pages' scale is fixed, taking into account the scale of the root one.
  - The clearing of the "Protocol" primitive's tail is fixed.

#### Optimization and performance improvement :

- *SYS*:
  - The mechanism of the "ResString" object resource is replaced from the RW-locks with mutex in order to improve performance and reduce memory consumption.
  - *TConfig*: The object of the configuration item (TCfg) is based on TVariant. An access to the "TCfg" object is made clear, which allows to use links without the basic types.
  - The mechanism for storing child nodes in the map wit the key "string" is changed to the key type "const char\*". All nodes are adapted to the constancy of the identifier.
- *Archive.FSArch*: The processing of the values' archive index table is significantly accelerated through the use of fast algorithm for computing the number of bits in a 32-bit integer.
- *UI.WebCfgD*:
  - The resizing of icons on the server side is added to reduce traffic.
  - The group request of the page's contents is used to greatly improve the performance on slow and high latency channels.
- *UI.QTCfg*:
  - The group request to the control interface is used for performance improvement on slow and high latency connections.
  - The multiple query of selective table columns elements is excluded.
- *UI.Vision*: The rendering performance of the fills of the "ElFigure" primitive is improved.
- *UI.VCAEngine*:
  - The measures to increase the speed of project's session initiation are implemented:
    - The initialization of the basic attributes because of their subsequent inheritance is skipped.
    - The translation of certain messages of the runtime session is removed.
    - The enabling of only the necessary pages at startup of the session is implemented. Other pages of the running session are enabled when accessed. This has greatly increase the speed of session's startup, as well as reduce the dependence of the starting speed on the complexity of the visualization project.
  - Implementation of the archive mode of the "Document" primitive is changed to direct work with the database, which let to expand the depth of the archive up to the 1000000 documents, and without prejudice to the RAM.

### 3 Improvement and stabilization of the graphics subsystem

The notable work was made within the graphics system, namely in the VCA engine modules *UI.VCAEngine*, visualizers *UI.Vision* and *UI.WebVision*, as well as in the configurators. The changes were provided for stabilization, the optimization of memory consumption and improvement of user's properties.

Improvements of the graphics subsystem:

- *The API of syntax highlighting is added.* On the basis of this API the syntax highlighting is added to:
  - *DAQ*:
    - *BlockCalc*: Syntax highlighting in description of the parameters' attributes.
    - *JavaLikeCalc*: Syntax highlighting for the compiler of the *JavaLikeCalc.JavaScript* language, as well as for all text fields using the language inside the module.
    - *ModBus, OPC-UA*: syntax highlighting of the attributes' list.
  - *Protocol.UserProtocol*: syntax highlighting to the text of the protocol's procedures.
  - *UI*:
    - *QTCfg*: multilevel syntax highlighting displaying, transmitted from the data model in the form of the rules of regular expressions.
    - *VCAEngine*: syntax highlighting for text attributes of the widget's primitives for which this is necessary.
    - *Vision*: syntax highlighting for text attributes of the widget's primitives, as well as for the texts of the widgets' programs.
    - *WebUser*: syntax highlighting in the procedures for constructing the Web-based interface.
- *UI.QTStarter*:
  - Adaptation to the ARM.
  - Font message in "Splash" is fixed at 10 pixels to ensure a uniform display.
  - The OpenSCADA stop is implemented in the case of closure the last dialog box of the QT UI-modules selection .
  - The forwarding of the QT4 command line options from the OpenSCADA is done (eg, *--qws* and *--style* to *-qws* and *-style*).
- *UI.QTCfg*:
  - The ability to manual change the height of the text and table fields is added. Change is made by grasping and dragging the bottom edge of the field.
  - Adaptation to the ARM.
  - Group request "CntrReqs" is used to query the page's context.
  - The restriction on the height of the rows of tables at half the height of the table is added.
  - The limit on the size of the input dialogue is added, and the adaptive determination of its size on its contents is added.
  - The fixed style "StyledPanel" is set for the elements, available to adjust their height by the user for the work elsewhere.
  - The function of full text editing of table's cells is added.
  - Playback engine of the syntax highlight rules is completely rewritten to simplify it and to avoid hidden errors.
- *UI.{VCAEngine, Vision, WebVision}*:
  - The addressing abroad the array's (vector's) size in the construction of trends of the "Diagram" primitive is fixed.
  - The width property of trend of the "Diagram" primitive is added and implemented.
  - The periodic inspection of the widgets' tree of the page is added in order to detect the fact of widgets' removing by requesting the complete list of widgets.
  - The display of trends, graphics group of the "Diagram" primitive in the scale of values is implemented in the case of differences between the scales of all graphs not more than 20%.
  - The ability to select the confirmation mode for the items of the line and text editing of the primitive "FormEl".
  - The possibility of loosing of some changes in the model, when displayed in the visualizers is fixed.

- The property of the values' count of the one pixel in the trends of the "Diagram" primitive is added and implemented in order to control granularity of export to CSV etc.
- The logarithmic scale of values for the trends of "Diagram" is implemented.
- The storage of aspect ratio property of the main page during the maximization of the main window of the project's session execution is added and implemented.
- The new type "Full Video" is added in the "Media" primitive. To implement the "Phonon" media engine in the *UI.Vision*.
- The field of the archive's properties "prm{X}prop" is added in the "Diagram" primitive, which is filled with the value of the available depth and frequency of the archive.
- *UI.VCAEngine*:
  - The specific properties of the attributes of primitives are added, such as help and syntax highlighting.
  - The enabling of project's session pages when it is really necessary is implemented in order to speed up the startup and optimize memory usage.
  - The "Document" primitive is remade for keeping the archives fully in the database, as well as the function of user API to access the elements of the archive is added.
  - The periodic (30 minutes) check and the closing of the unused (orphaned) projects' sessions are added.
  - The previous one (controversial) rollback mechanism of nested widgets deleting is removed and it is replaced by the full-fledged rollback mechanism in the *UI.Vision* visualizer.
  - Optimization of memory usage by the session's elements up to 20% is implemented.
  - The support of the first, out of date, version of the database structure of the visual area is removed.
  - The restriction on the time of the document's generation in the "Document" primitive in 5 seconds is added.
  - The support of the attribute's type "Object" and "Text" is added.
  - The forced initialization of new user's attributes in EVAL is disabled.
  - The display of help for the widget's attributes is added.
  - The "Document" primitive is switched to XML parsing in order to fully preserve the XHTML markup.
  - The display of the correct status of the links in the form of "(+)" at the end of the line is added.
  - The redirection to the above page of raw events is added. This allows you to globally handle events at the very top page.
  - The ability to make the links between widgets of "wdg:" type in the form of absolute and relative ones.
  - The support of the "arh:" link type is added.
  - The support of the data source parameters' object type is added.
- *UI.Vision*:
  - The display of help for the widget's attributes is added.
  - Adaptation to the ARM.
  - The display of message about loss of connection with the visualization server and its recovery is added for remote visualization.
  - The separate full-text editor for text attributes is added.
  - The implementation of the full-fledged changes undo and redo in visual editing of widgets is added.
  - Improvements in the sensitivity and updating the changes in the project's session: the rendering performance of the "ElFigure" primitives is optimized, handling of the changes in the "geomZ" level of the widget is added.
  - The export to CSV format from the "Diagram" and "Document" primitives is added.
  - The limit on the size of the input dialogs is added and an adaptive determination of their sizes by their contents is added.
  - The rules' execution mechanism for syntax highlighting is completely rewritten.
  - The new session creation after the restoration of remote connection and absence of the previous session is added.

- Window background for the editable widget is set to “QT::Dense7Pattern” template to eliminate the possible of overlapping by color.
- The event handling of the widget's selecting when press in the scroll area, but outside of the widget is added.
- "Elementary figure" primitive:
  - The dialog to set properties for the separately selected figures is added.
  - The reorganization of the operations of the Properties dialog and context menu with dynamic and static properties is completed.
  - In the process of addition the figure its rendering when moving the mouse cursor is implemented.
- *UI.WebVision*:
  - The support for “gif” and “jpg” images in the elements of the "Elementary figure" primitive is added.
  - The handling of nested frames scale is fixed, as well as the mechanism of formation-enabling of scroll is revised.
  - The function of the images conversion on the server's side is added. Conversion function is used for resizing and discoloration the images of inactive buttons.
  - The opening of small windows of the nested pages as the DIV-blocks is implemented, it solves the problem of lock and the long opening of the external windows in many browsers.
  - The main page of the interface is centered in the browser's window.
- *UI.WebCfgD*:
  - The brightness of decolorized-passive buttons is reduced.
  - All dialogs windows are implemented as DIV-blocks in order to avoid problems and delays in opening external windows in different browsers.
  - The group request the page's contents is used for the significant improvement of the performance on slow and high latency channels.
  - The information in the status bar about the current user and the possibility of its change are added.
  - The adaptation to work in full screen width is made.

## 4 Formation, expansion and stabilization of the user programming API

It was continued the development of object API of the user programming, which provides the integration of user's functions in the object's tree of OpenSCADA. In addition, a number of changes were made to the existing library of the user's API functions.

In particular the following changes were made:

- *SYS*:
  - User's API for the TArrayObj object is extended for the possibility of using object properties if the properties are not the numbers.
  - In the object tree of OpenSCADA of the user's API the *SYS.Security.access()* function is added to control user access to OpenSCADA resources with the necessary privileges.
  - The functions to work with files are added: *SYS.fileRead()* and *SYS.fileWrite()*.
  - The *getElementBy()* function is added to the XMLNodeObj object's API to search for embedded nodes by attribute's value.
  - The function of encoding the text between different character encodings is added.
  - The *NodeObj.nodePath()* function is added.
- *SYS, DAQ.JavaLikeCalc*: The implementation of the support for regular expressions of user's API, based on "PCRE", through the "TRegExp" object is added. The interface of functions *RegExp.exec()* and *RegExp.test()* is implemented in the standard for the "JavaScript" language. The "**new RegExp()**" initialization of the "TRegExp" object is implemented for the *DAQ.JavaLikeCalc*. The additional functions using "RegExp": *int search()*; *Array match()*; *Array split(RegExp pat, int limit)*; *string replace(string substr, string str)*; *string replace(RegExp pat, string str)* are implemented for *DAQ.JavaLikeCalc*.
- *TConfig, TBD, TTransportIn, TTransportOut, TUser, TGroup, TPrmTmpLib, TPrmTempl*,

*TParamContr*, *TController*, *TVArchive*, *TVArchivator*, *TMArchivator*: The following functions of the user API are added: *cfg()* and *cfgSet()* to access the configuration of the objects (stored in the database).

- *TFunction*, *DAQ.JavaLikeCalc*: The user's API object "TFuncArgObj" is added for function arguments get.
- *DB*: The following functions of the user's API are added: *fieldStruct()*, *fieldSeek()*, *fieldGet()*, *fieldSet()* and *fieldDel()* for the table object to provide the user access to the database through the database interface OpenSCADA.
- *DAQ*:
  - The *TController::alarmSet()* function is added to generate the typical alarms in the controller's object of the "Data acquisition" subsystem's modules.
  - The *SYS.DAQ["Modul"]["Controller"]["Parameter"]["Attribute"].arch()* function is added to directly access the object's archive associated with the attribute of the parameter.
  - The following functions of the user API are added: *enable()* and *start()* for the direct monitoring of the controller's object.
  - *JavaLikeCalc*:
    - The built-in function *typeof()* to check the value's type is added.
    - The *isEVal()* function for the "null" data type is added, which always returns "true".
    - The *call()* function of the user API is added to the objects of the library functions.
  - *ModBus*: The *messIO()* function of the user API is added in the "ModBus" controller's object in order to provide an opportunity to send custom ModBus-queries directly from the template of the parameter.
- *DAQ.JavaLikeCalc*, *Special.{FLibComplex1, FLibMath, FLibSYS}*: The function for the dynamic call of the library functions is added: *SYS.DAQ.JavaLikeCalc["lib\_{Lib}"].funcId(prms, ...)*.
- *Archive*: The following functions of the user API are added: *status()*, *end()* and *begin()* for access to the state and properties of an object of the messages' archiver.
- *UI.VCAEngine*: The *wdgAt()* function of the user's API is added for easy connection to the widget, both in absolute and relative path.

## 5 System-wide extensions

There have been made the significant changes and extensions to the system-wide API of the OpenSCADA system with the purpose of the general stabilization and in view of the general measures for cleaning the source code and reorganizing the sources' tree:

- *SYS*:
  - The new structure of the sources tree is implemented.
  - All the resources of the project are moved to the separate archive.
  - To create sources' and resources' archives it can be used the command "\$ make dist-lzma" and "\$ make dist-reslzma".
  - Documentation is made the static for modules and is divided by languages. The separate package is formed for each documentation language. In the documentation of each language it is created an index file for the entire list of local and network documentation. The total list of local documentation is greatly expanded with new documents, as well as with the supported network documentation.
  - The packages of librarian databases are added: "\*-LibDB.Main" and "\*-LibDB.VCA", they are placed to "data/LibsDB".
  - The packages of the "AGLKS" and "Boiler" models are added and placed to the "data/ModelsDB" instead of the "DemoDB" package.
  - Configuration files to build RPM and DEB packages are updated.
  - Dependence on the library PCRECPP is globally replaced by PCRE.
  - The new function *TSYS::time2str()* is added to convert time intervals into a string like "1hour 23min 10sec". This function is used everywhere.
  - The implementation of own XML-parser is added, instead of Expat. The own XML-parser solves the problem of supporting different character sets other than UTF, as well as

the problem of safe handling the situation of the characters presence that are not valid for the current character encoding.

- *TMess*: Common control interface messages are moved into one *TMess::lab\** function.
- The storage object of the string with the "ResString" resource is significantly expanded for the clear conversion from/to "std::string type".
- The ability to save to the configuration file is added:
  - The database prefix "<cfg>" is added for presentation of the configuration file as a source of load/store configuration.
  - The *TDBS::dataSeek()* and *TDBS::dataDel()* functions are updated for the strict processing the configuration file.
  - The *chkSelDB()* function is adapted to support the record to the configuration file.
  - The storage of large and multi-line field values in the text box of the individual tag of the configuration file is added.
- The "long long" type, usually a 64-bit integer is replaced everywhere by the "int64\_t" on all architectures.
- The objects of the OpenSCADA threads are rigidly linked with their thread. The load and periodic threads' calls statistics is added.
- The possibility of checking the loaded objects for their absence in the database and deleting is added. Check function is activated only in the case of direct loading from the database.
- The *perSYSCall()* system periodic function for the call of subsystems' and their modules' objects is added. It is used for service purposes of unloaded and rare task of the modules and subsystems.
- The test function of the expression by the template is moved to the "TRegExp" object.
- *XMLNode*:
  - The code of "Expat" XML-parser using is completely removed.
  - The ability to load text parts of the tag to the separate specific tags in full mode is added. It is designed for full storage the configuration of the tag.
  - The ability to load and store the comments blocks into the separate specific tags in full mode is added.
- The built-in function of enabling the generation of a memory dump (removal of restrictions on file size) by setting the start OpenSCADA *--CoreDumpAllow* option is added.
- The automatic memory dump file processing procedure (the creation of the stack reversal - backtrace) is added to all the starting scripts of OpenSCADA. A memory dump file is expected to be in the working directory as a "core".
- The ability to include (embed) modules to the kernel library of OpenSCADA is added.
- The version of the module API of all subsystems is increased to 6.
- *TConfig*, *TValFunc*, *TVal*: The storage the string configuration fields is moved to the "ResString" object of the resource string , and then it is returned with the use of internal resource.
- *TVariant*: The storage mechanism for the string is changed in the storage of the pointer to an array of "union" with an adaptive mechanism for memory allocation (simple one, up to 7 characters is stored statically, but for the longer ones the more memory is allocated dynamically).
- The building system has been significantly updated:
  - The check and disabling of the "Iconv" and "Intl" (I18N) API using is added.
  - The automatic detection and enabling of the *-enable-CrossCompile* configuration parameter for cross-compilation environment is added.
  - The *-enable-CoreLibStatic* configuration option is added to disable the OpenSCADA core shared library build and static linking of the call-start program.
  - The summary configuration table is added.
  - Moving the I18N processing code to the I18N.mk file and including it directly to the Make-files, if it is necessary, on the last stage of the configuration is added.
  - The general macro *AX\_MOD\_EN()* is added for the unified modules control.
  - The installation of the included files (headers) is added.

- The openscada.pc file to check the OpenSCADA configuration in the construction of external modules is added.
  - The global version of the OpenSCADA source files version is changed to 0.8.0 and the OpenSCADA core library version is changed to 2.0.0.
- Implementation of the manipulation with objects through the parameters' attributes:
  - The overall threading safety for the "TVarObj", "TArrayObj" and "XMLNodeObj" user's API objects is implemented: The automatic removal of the object in the "AutoHD" at the "true" AHDDisConnect() function's return.
  - The reverse serialization (parsing of the object's XML representation) is added for the "TVarObj", "TArrayObj" and "XMLNodeObj" objects. It is enabled for the "TVariant", "TVal", "TValFunc", "UI.VCAEngine.Widget", that allows you to save and load a value of the object type to/from the database.
  - *TVal, DAQ, {BlockCalc, JavaLikeCalc, LogicLev, ModBus, Siemens}, UI.VCAEngine*: The support of the a "Object" value's type is added. It is enabled for general use.
- *DOC*:
  - Check and repair of the English texts in the OpenSCADA core are made. The translations for the OpenSCADA core to German, Russian and Ukrainian languages are updated.
  - Check and repair of the English texts in the OpenSCADA modules are made. The translations for the OpenSCADA modules to German, Russian and Ukrainian languages are updated.
  - The all key documents in accessible languages are updated: openscada.pdf, build.pdf, properties.pdf, "OpenSCADA API", "About OpenSCADA", "TechApp Library", "Quick Start", "WLib Main", "WLib Mnemo Elements".
- *Libraries databases*:
  - The library of the main visual elements is appended by:
    - the frames of the implementation editing and execution of "Prescriptions" - "User's programs";
    - selection and configuration dialog of the the "Graphics group" frame parameter;
    - some improvements are added to the "Graphics group" frame for the selection of the parameters by the user and for other purposes;
    - the horizontal scroll bar is added to the "Graphics group" frame for the comfortable viewing the archives.
  - The [library of the electrical elements](#) is added to the kit of libraries package.
  - The frame of the main page is appended by the mechanism of "playing" the models of technological processes (TP).
    - The [industrial boiler unit\(RUS\)](#) model is translated to English and Ukrainian languages.
- *DB*: The time limit of 5 seconds and the field of specifying the starting offset are added for the request of the large tables' contents.
- *Transport*:
  - The function *TTransportS::traf2str()* is added to convert the value of the traffic's counter to the string like "12.5KiB".
  - Types of formats for outgoing requests are expanded with versions of the text with a different end of line: LF, CR, CR/LF.
- *DAQ*:
  - The ability to select the type of the "object" template's attribute is added.
  - *DiamondBoards, ModBus, LogicLev*: The ability to change the type of the parameter for the disabled parameters of multi-type modules of "Data acquisition" subsystem is added.
  - *LogicLev, ModBus, Siemens*:
    - In the case of the scheduled implementation the time after the last calculation is written in the "f\_freq" attribute as the negative one.
    - The initialization and stop calls are added at the start/stop of the parameter.
    - The update of the boolean parameter's attributes when changing the type, the "ReadOnly" flag or the name in the template.
- *TArchives*:

- The ability to use regular expressions when searching by category with the help of `"/match/"` is added.
- *DBArch*, *FSArch*: The time border when requesting messages from the archive is added.
- The link's address on the parameter's attribute is simplified to the "DAQMod.Cntr.Prm.attr" form. Support of the old address is stored.

## 6 Publication of OpenSCADA solutions

During the work on this version the solutions, based on the OpenSCADA, have been presented at various conferences and exhibitions:

- [FOSS Sea 2011](#) — the report about the solutions, based on the OpenSCADA was read at the conference and in the hall there was made the mini-stand with the live solutions, based on OpenSCADA.
- [International Innovation Forum 2011](#) — at the stand of "DIYA" Ltd and Dneprodzerzhinsk State University (DSTU) the equipment and materials on the solution, based on OpenSCADA were presented.
- [10 Ukrainian Conference for Free Software Developers and Users](#) — the report about the solutions, based on the OpenSCADA was read at the conference and in the hall there was made the mini-stand with the live solutions, based on OpenSCADA.
- [Qt Developer Days 2011](#) in Munich and San Francisco: there were made the demonstration of OpenSCADA solutions [Dynamic model of the DMKD multifuel steam boiler unit №9](#) on the PC and [The dynamic model of real-time of the Anastasievskaja GLKS](#) on Nokia N9 in the Qt Demo Pavilion.

## Conclusion

During the creation the new stable 0.8.0 release for industrial use with long-term support a lot of work to stabilize, enhance the functionality and adapt for running on the alternative hardware platforms (on the ARM architecture) is done. All this is in general make it possible to extend the sphere of OpenSCADA using on all levels of automation from the PLC, SCADA-systems server to the distributed visualization systems.

The following people took an active part in the appearance of the new industrial 0.8.0 version of the OpenSCADA system:

- [Roman Savochenko](#): The bulk of the work on developing, documenting, and testing.
- [Maxim Lysenko](#): Maintenance and stabilization of the "Elementary figure" primitive of the visual control area (VCA), as well as translation and update much of the documentation in English.
- [Irina Popkova](#): Translation and update of the OpenSCADA system interface and its modules in German language.
- [Almaz Karimov](#): Expansion of the DCON protocol module support and active testing.
- [Aleksey Popkov](#): Building packages for Fedora-based distributions.
- Many other OpenSCADA users through comprehensive testing.

Further development efforts will be directed to the software multiplatform implementation, and expanding of functionality by creating new interfaces support modules, including the advanced development scheme, with the implementation costs distribution for the sold copies (for the modules, which interact with the commercial systems and interfaces). In order to implement a centralized distribution of solutions, services and enhancements based on OpenSCADA the creation of the store is planned within the main information OpenSCADA resource (<http://oscada.org>).

## Links

Testing protocol of production release: <http://wiki.oscada.org/Works/Tests/release080>